

# ENTWICKLUNG EINER WEBANWENDUNG (WORKSHOPPY) ZUR DURCHFÜHRUNG VON WORKSHOPS IN ECHTZEIT

## **Abschlussarbeit**

zur Erlangung des akademischen Grades  
Bachelor of Science (B.Sc)

an der

Hochschule für Technik und Wirtschaft Berlin  
Fachbereich 4  
Studiengang Angewandte Informatik

vorgelegt von

**Alongkorn Kiatmontri**

(eingereicht am )

**Erstprüfer:** Herr Prof. Jung, Th.

**Zweitprüfer:** Herr Andreas Flack (LB)

## **Eidesstattliche Erklärung**

Hiermit versichere ich, die vorliegende Abschlussarbeit selbstständig und nur unter Verwendung der von mir angegebenen Quellen und Hilfsmittel verfasst zu haben. Sowohl inhaltlich als auch wörtlich entnommene Inhalte wurden als solche kenntlich gemacht. Die Arbeit hat in dieser oder vergleichbarer Form noch keinem anderem Prüfungsgremium vorgelegen.

Datum: \_\_\_\_\_ Unterschrift: \_\_\_\_\_

## **Danksagungen**

## **Zusammenfassung / Abstract**

### **Abstract**

# Inhaltsverzeichnis

|   |           |
|---|-----------|
| Eidesstattliche Erklärung . . . . .             | 2         |
| Danksagungen . . . . .                          | 3         |
| Zusammenfassung / Abstract . . . . .            | 4         |
| <b>1 Einleitung</b>                             | <b>9</b>  |
| 1.1 Motivation . . . . .                        | 9         |
| 1.2 Zielsetzung und Aufbau der Arbeit . . . . . | 10        |
| <b>2 Grundlagen</b>                             | <b>11</b> |
| 2.1 Grundlegende Begriffe . . . . .             | 11        |
| 2.1.1 Workshop . . . . .                        | 11        |
| 2.1.2 Brainstorming . . . . .                   | 13        |
| 2.1.3 TCP/IP . . . . .                          | 14        |
| 2.1.4 HTTP . . . . .                            | 16        |
| 2.1.5 Ablauf einer HTTP-Verbindung . . . . .    | 17        |
| 2.1.6 AJAX . . . . .                            | 18        |
| 2.1.7 Echtzeit . . . . .                        | 19        |
| 2.1.8 WebSocket . . . . .                       | 22        |
| 2.1.9 jQuery . . . . .                          | 25        |
| 2.2 Responsive Webdesign . . . . .              | 26        |
| 2.3 Web 2.0 . . . . .                           | 27        |
| 2.4 Rich Internet Applications . . . . .        | 30        |
| 2.5 Thick Client und Thin Client . . . . .      | 32        |
| <b>3 Analyse</b>                                | <b>33</b> |
| 3.1 Stand der Technik . . . . .                 | 33        |
| 3.1.1 IdeaBoardz . . . . .                      | 33        |
| 3.1.2 Miro-RealtimeBoard . . . . .              | 36        |
| <b>4 Design</b>                                 | <b>38</b> |
| <b>5 Implementierung</b>                        | <b>39</b> |

|                             |           |
|-----------------------------|-----------|
| <b>6 Fazit und Ausblick</b> | <b>40</b> |
| <b>Literaturverzeichnis</b> | <b>41</b> |

# Abbildungsverzeichnis

|      |   |    |
|------|---|----|
| 2.1  | Aufbau eines Datagramms . . . . .   | 14 |
| 2.2  | Zeitüberschreitung und erneute Übertragung bei Verlust eines Pakets . . . . . | 15 |
| 2.3  | Die wichtigsten Protokolle im TCP/IP-Referenzmodell . . . . .                 | 15 |
| 2.4  | Das Client-Server-Modell . . . . .  | 16 |
| 2.5  | Aufbau einer URL . . . . .  | 17 |
| 2.6  | Klassisches HTTP Request-Response-Paradigma nach [Wöhr, 2004] . . . . .       | 17 |
| 2.7  | synchrone und asynchrone Kommunikation . . . . .                              | 19 |
| 2.8  | Polling . . . . .   | 21 |
| 2.9  | LongPolling . . . . .   | 21 |
| 2.10 | Beispiel einer Client-Handshake-Anfrage . . . . .                             | 23 |
| 2.11 | WebSocket-Server Handshake . . . . .  | 23 |
| 2.12 | Das WebSocket-Handshake . . . . .   | 24 |
| 2.13 | DOM - Elementenbaum einer Webseite . . . . .                                  | 25 |
| 2.14 | Das Konzept Web 2.0 nach einer Brainstorming-Sitzung . . . . .                | 28 |
| 2.15 | Die Bedeutung von Web 2.0 . . . . .   | 28 |
| 2.16 | Kommunikationsmodell einer klassischen Webanwendung . . . . .                 | 30 |
| 2.17 | Kommunikationsmodell einer RIA am Beispiel einer AJAX-Anwendung . . . . .     | 31 |
| 2.18 | Vergleich der Desktopanwendung, klassischen Webanwendung und RIAs . . . . .   | 32 |
| 3.1  | Erstellen eines eigenen IdeaBoards. . . . .                                   | 34 |
| 3.2  | Darstellung von Sektionen . . . . .   | 34 |
| 3.3  | Darstellung einer der Sektionen . . . . .                                     | 35 |
| 3.4  | Zusammenführen und Zuordnen von Daten . . . . .                               | 35 |
| 3.5  | Realisieren eines Brainstormings mit Hilfe von Miro . . . . .                 | 37 |

# **Tabellenverzeichnis**



# 1 Einleitung

Im ersten Kapitelabschnitt der Bachelorarbeit, wird auf die Motivation und die Zielsetzung eingegangen. Zusätzlich wird ein Überblick über den Aufbau der Arbeit aufgezeigt.

## 1.1 Motivation

Beim Suchen und Finden von Lösungen, ungewöhnlichen Geschäftsideen, Innovationen oder um einzelne Projekte erfolgreicher zu machen, bereichert viele Menschen der Begriff Kreativität. Um die Kreativität zu fördern, braucht es Kreativitätstechniken, die dabei helfen, Ideen zu generieren und Einfälle zu sammeln.

Der Klassiker und eine der bekannteste unter allen Kreativitätstechniken ist das klassische Brainstorming. Sie wurde vom Amerikaner Alex Faickney Osborn erfunden und von Charles Hutchison Clark zur Ideenfindung innerhalb von Gruppen weiterentwickelt. (vgl. [van Aerssen, oJ]) „Er benannte das Brainstorming nach der Idee dieser Methode, nämlich using the brain to storm a problem (wörtlich: Das Gehirn verwenden zum Sturm auf ein Problem).“ [Holzer, 2012]

Die Kreativitätstechnik Brainstorming gilt als eine der beliebtesten Methoden zur Ideenfindung und -sammmlung neuen Geschäftsideen, Ideen für ein Projekt/Produkt oder auch zu einer vorhandenen bzw. gegebenen Problemstellung.

Ziel des Brainstormings ist es, Denkblockaden auf der Suche nach neuen Ideen zu beenden. Diese Kreativitätstechnik wird häufig in Seminaren und Workshops angewendet, um die Gruppenarbeit effektiver und effizienter zu gestalten. Bei einer Brainstorming-Sitzung in einem Workshop kann jeder Teilnehmer auf die Ideen des anderen aufbauen und anknüpfen. Dadurch werden die Teilnehmer gegenseitig durch Ihre Ideen zu neuen Ideen angeregt, wodurch mehr Ergebnisse, als tatsächlich gebraucht, produziert werden.

Eine häufig angewendete Methodik für die Ausarbeitung des Brainstormings in den Workshops ist es, sich Karteikarten oder Notizzettel zu nehmen, seine Ideen und Gedanken darauf zu schreiben und an eine Pinnwand (Flipchart, Whiteboard) anzubringen. Haben alle Teilnehmer Ihre Karteikarten an der Pinnwand angebracht, wird anschließend analysiert und darüber diskutiert. Am Ende der Besprechung werden die gesammelten Daten bewertet und anschließend von dem Moderator dokumentiert. Mit herkömmlichen analogen Workshops bedeutet das für den Moderator, dass er die Karteikarten auf der

Pinnwand abtippen oder abfotografieren muss, um eine Dokumentation erstellen zu können. Da wir uns heutzutage in einem digitalen Zeitalter befinden und uns dieser neuen Welt nicht mehr entziehen können, gilt es, diesen Wandel als Chance zu begreifen, solche analogen Workshops zu digitalisieren, um dem Moderator eine Möglichkeit anzubieten, die Daten digital zusammenzufassen.

## **1.2 Zielsetzung und Aufbau der Arbeit**

Im Rahmen dieser Bachelorarbeit soll eine dynamische Webanwendung (Workshoppy) zur Durchführung von Workshops in Echtzeit entwickelt werden, die das klassische Brainstorming digitalisieren und effektiver machen soll. Die Webanwendung soll künftig in den Workshops genutzt werden und muss die Funktionen bieten, welche mehrere Personen (Teilnehmer) über ihre Endgeräte (Smartphone, Laptop oder Tablet) ihre Ideen abgeben können. Dabei werden die eingebrachten Ideen der Teilnehmer in Echtzeit auf einer großen Leinwand (Beamer) präsentiert. Der Moderator soll anschließend die Möglichkeit erhalten, die Ergebnisse digital zusammenzufassen. Die Zusammenfassung soll auch als PDF-Datei exportiert werden können. Bei der Konzeption der Webanwendung ist zu beachten, dass eine benutzerfreundliche Darstellung für die Anwender gewährleistet ist.

Die vorliegende Arbeit ist wie folgt aufgebaut: Das Kapitel 2 stellt vorab ein Überblick über einige grundlegende Begriffe vor. Der Begriff Responsive Webdesign, AJAX-Technologie und Rich Internet Applications (RIA) werden besprochen. Anschließend wird der Thin Client und Thick Client beschrieben. Das Kapitel 3 beschäftigt sich zunächst mit dem Stand der Technik. Die Anforderung zur Webanwendung wird dabei analysiert und konzipiert. In diesem Kapitel werden vor allem die funktionale, nicht-funktionale Anforderungen sowie die Muss- und Kann- Anforderungen ermittelt. Aufbauend auf den Ergebnissen der Anforderungsanalyse erfolgt in Kapitel 4 eine ausführliche Beschreibung über den Entwurf der Benutzeroberfläche (GUI) der Webanwendung. Danach wird das Design der GUI entworfen. Im Kapitel 5 wird zunächst die zu verwendenden Webentwicklungswerkzeuge vorgestellt. Anschließend beschäftigt sich dieses Kapitel hauptsächlich mit der Implementierung der Webanwendung. Zum Schluss wird es im Kapitel 6 die erarbeiteten Ergebnisse zusammengefasst, sowie Ideen für zukünftigen Erweiterungen der entwickelten Webanwendung diskutiert.

## 2 Grundlagen

Dieses Kapitel behandelt die für diese Arbeit nötigen Grundlagen. Zunächst wird ein Überblick über grundlegende Begriffe vorgestellt. Das Responsive Webdesign und einige Merkmale für eine responsive Webseite in Bezug auf die zu entwickelnde Webanwendung werden erläutert. Dann werden das Web 2.0 und der entstandene Ausdruck *Rich Internet Application* beschrieben. Anschließend gibt es die Unterschiede zwischen Thin Client und Thick Client.

### 2.1 Grundlegende Begriffe

#### 2.1.1 Workshop

Workshop<sup>1</sup> ist eine Veranstaltung, bei der sich eine bestimmte Anzahl von Personen teilnimmt, um außerhalb der Routinearbeit Fragen, Probleme und Themen zu bearbeiten. Jeder Workshop wird von einem Moderator geleitet. Bei größeren Gruppen (mehr als 15 Teilnehmer) ist der Einsatz von weiteren Moderatoren zu empfehlen. Die Teilnehmer handeln es sich in der Regel um Spezialisten oder Betroffene, die Ihr Fachwissen zu der behandelten Aufgabe einfließen lassen. Das Ziel ist dabei: Lösungsvorschläge für Aufgaben- oder Problemstellung zu generieren und Maßnahmenplan für die Umsetzung zu entwickeln.

Der Moderator ist der aktiver Dienstleister der Gruppe. Er ist für die Vorbereitung sowie Organisation verantwortlich und soll die Gruppe am Ende zum Ziel führen. Seine Aufgaben bestehen unter anderem, Fragestellung gezielt zu formulieren, den Ablaufplan zu erstellen, Denkprozesse anzuleiten, Zeitplan einzuhalten und Ergebnisse zu dokumentieren. Er muss außerdem die stille Teilnehmer aktivieren sowie die dominante bremsen und darauf achten, dass die Gruppe bei Diskussionsrunden das Ziel nicht aus den Augen verliert.

Nach Ansicht des Autors [phil. Tomas Bohinc, 2016] können Workshops in folgenden Projektphasen eingesetzt werden:

- Kick-Off-Veranstaltung
- Projektplanung-Prozess
- Problemlösung

---

<sup>1</sup>bedeutet so viel wie „Arbeitskreis oder -gruppe“ .

- Entscheidungsfindung
- Informationsaustausch
- Teamentwicklung
- Scrum
- Projektabschluss

Die Gestaltung von Workshops spielt bei der Qualität der Ergebnisse eine große Rolle. Bei einem unstrukturierten Workshop kann dazu führen, dass er keine Motivation bei den Teilnehmer erregt, um sich an dem Workshop einzubringen und Ergebnisse zu erarbeiten. Um dagegen vorzugehen, können der Moderator je nach Dauer des Workshops folgenden kreative Workshop-Methoden anwenden, um Workshops effektiv und interaktiv zu gestalten:

- World Cafe
- Open Space
- Six Thinking Hats
- Fishbowl
- Lego Serious Play

Die genauen Beschreibungen zu den oben genannten Workshop-Methoden können im Blogpost von [Chornaya, 2018] verfolgt werden.

Wenn es darum geht, neue Ideen für Problemlösung, neue Produkten, neue Geschäftsideen oder Innovationen zu erzeugen, werden Kreativitätstechniken eingesetzt. Denn durch Kreativität werden Ideen generiert. Viele moderne Kreativitätstechniken haben sich im Laufe der Jahre etabliert. Dem Moderator steht deshalb eine Vielzahl von Kreativitätstechniken zur Verfügung. Der Klassiker und eine der beliebtesten unter allen Kreativitätstechniken ist wie bereits im Unterkapitel 1.1 erwähnt, das klassische Brainstorming. Da die vorliegende Arbeit eine Webanwendung zur Durchführung von Workshops behandelt, die das klassische Brainstorming digitalisieren soll, werde ich deshalb nicht auf die anderen vorhandenen Kreativitätstechniken eingehen.

### 2.1.2 Brainstorming

Wie bereits im Unterkapitel 1.1 benannt, werden beim Brainstorming anhand eines konkreten Thema bzw. Problems Ideen, Einfälle und Vorschläge gesammelt. Es kommt dabei nicht auf die Qualität der Ideen an, sondern zunächst, dass möglichst viele Ideen generiert werden. Beim Brainstorming zählt die Quantität vor Qualität. Die Teilnehmer in der Gruppe sollen ihre Gedanken öffentlich frei äußern. Durch diesen öffentlichen Austausch, können mehr Ergebnisse produziert werden.

Die Gruppengröße bei einer Brainstorming-Sitzung sollte nicht zu groß und zu klein sein. „Je nach Fachliteratur ist von Gruppengröße von 5 bis maximal 20 Personen die Rede“. [Holzer, 2012]

Nach [Reich, 2007] läuft eine Brainstorming-Sitzung in folgenden Phasen ab:

- **Vorbereitung:**  
Der Moderator stellt in dieser Phase die zu behandelte Frage und die Regeln vor. Bei Notwendigkeit kann ein oder mehrere Protokollant/en bestimmt werden.
- **Ideen sammeln:**  
Die Teilnehmer dürfen Ideen und Vorschläge frei äußern. Der Moderator muss in dieser Phase vor allem die stillere Teilnehmer motivieren und ermuntern. Die Kritik ist in dieser Phase untersagt. Die Ergebnisse werden dabei protokolliert. In der Regel werden die Ideen auf eine Notizzettel geschrieben und an die Wand gepinnt.
- **Zusammenfassung und Auswertung:**  
Das Brainstorming ist nun beendet und der Moderator wird die Gruppe zunächst die dokumentierte Ergebnisse präsentieren. Anschließend werden die Ideen gemeinsam mit der Gruppe ausgewertet, sortiert und geordnet. In dieser Phase ist Kritik erlaubt und darf geäußert werden. Am Ende dieser Phase soll eine Liste mit den gut bewerteten Ideen und Vorschläge entstehen.
- **Nachbereitung:**  
Ein Brainstorming fördert nur die Kreativität. Die Vorschlägen müssen danach umgesetzt und realisiert werden. Sonst helfen die Ideen nicht, wenn nichts daraus gemacht wird.

Damit eine Brainstorming-Sitzung erfolgreich verlaufen ist, sollten dabei folgenden Regeln eingehalten werden:

- Unabhängig wie verrückt jede einzelne Idee ist, keine Kritik in der Sammlungsphase.
- Quantität vor Qualität, je mehr Ideen desto besser.
- Entwicklung oder Verbesserung von fremden Ideen ist willkommen.
- Lass die Fantasie freien Lauf. Ungewöhnliche Ideen sind erwünscht.

### 2.1.3 TCP/IP

Das Transmission Control Protocol (TCP) und das Internet Protocol (IP) bilden Grundlage für die gesamte Netzwerkkommunikation und legen demnach die grundlegende Technologien für das Internet dar.

TCP nutzt für die Übertragung der Datenpakete das Übertragungsprotokoll IP, welches zur Vermittlungsschicht im TCP/IP-Referenzmodell gehört. Die Aufgabe von IP-Protokoll ist, die Datenpakete an den richtigen Rechner im Netzwerk zu transportieren. Die Datenpakete sind nicht anderes als Datagramme. Ein IP-Datagramm enthält unter anderem die IP-Adresse des Absenders und des Empfängers sowie weitere spezifische Übertragungsparameter (**Abbildung 2.1**). Ob alle versendeten Datagramme erfolgreich beim Empfänger angekommen sind, kann das IP-Protokoll jedoch nicht sicherstellen. Solche Fehlerbehandlungen, wie z.B. ob Pakete beim Empfänger tatsächlich angekommen sind, stellt die Transportschicht, allem voran TCP, sicher. (vgl. [Karadeniz, oJ])

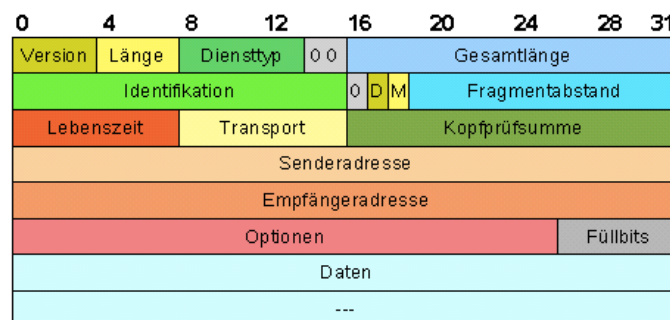


Abbildung 2.1: Aufbau eines Datagramms

**Quelle:** <http://einstein.informatik.uni-oldenburg.de/rechnernetze/diagramm.htm>

TCP ist eines der wichtigsten Protokolle der Transportschicht im TCP/IP-Referenzmodell und ist ein zuverlässiges, verbindungsorientiertes und paketvermittelndes Transportprotokoll, welches das Ziel hat, Datenverluste bei der Datenübertragung zu unterbinden, größere Datenmenge in kleinere Pakete zu zerlegen und die empfangene Datenpakete über Ports an den korrekten Anwendungen weiterzuleiten. Da sich TCP ein verbindungsorientiertes Protokoll handelt, definiert das TCP-Protokoll eine Ende-zu-Ende-Verbindung zwischen den Kommunikationspartnern im Netzwerk. (vgl. [o.V., 2019])

Das TCP-Protokoll verwendet dabei das Verfahren namens *Positive Acknowledgement (ACK) with Retransmission*<sup>2</sup>, um die Zuverlässigkeit der Datenübertragung sicherzustellen. Dies hat zu bedeuten, dass der Empfänger nach dem Erhalt der Daten dem Sender mit einer positiven Nachricht quittiert wird. Mit einer positiven Nachricht weiß der Sender, dass das Paket den Empfänger erreicht hat. Sollte von

<sup>2</sup>auf deutsch: positive Bestätigung mit erneuter Übertragung

seitens der Empfänger keine positive Nachricht kommt, wird das Senden solange wiederholt, bis eine positive Antwort beim Sender eingegangen ist (**Abbildung 2.2**). (vgl. [Holtkamp, 2001])

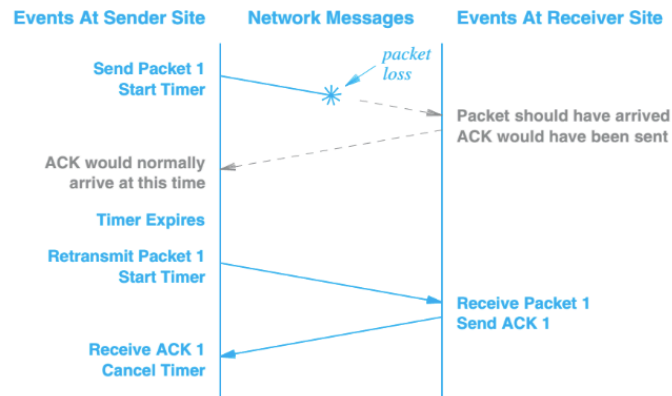


Abbildung 2.2: Zeitüberschreitung und erneute Übertragung bei Verlust eines Pakets

**Quelle:** [http://lemoncisco.blogspot.com/2014/06/internetnetworking-with-tcpip-notes\\_25.html](http://lemoncisco.blogspot.com/2014/06/internetnetworking-with-tcpip-notes_25.html)

Die folgende Abbildung (**Abbildung 2.3**) zeigt die wichtigsten Protokolle im TCP/IP-Referenzmodell. Über der TCP- und IP-Schicht im TCP/IP-Referenzmodell befindet sich die Anwendungsschicht. Diese Schicht beinhaltet alle Protokolle auf Anwendungsebene, die auf TCP oder UDP aufsetzen. Die Anwendungsschicht stellt den Anwendungsprogrammen Dienste zur Verfügung. Das bekannteste Protokoll auf der Anwendungsschicht ist wohl das Hypertext Transfer Protocol (HTTP), welches den Zugriff auf die Webseiten ermöglicht.

| Schicht   | DoD-Schichtmodell<br>theoretisches<br>Arbeitsmodell          | TCP/IP-<br>Protokollsammlung<br>praktische Ausführung                 |           | OSI-<br>Schicht Nr. |
|---|--|---|-----------|---------------------|
| 4<br>Process<br>Anwendungsschicht   | Anwendungen<br>und Nutzdaten<br>interaktive Protokolle       | HTTP, SMTP, FTP, DNS<br>POP3, IMAP, Telnet<br>SNMP, NTP               | Stream    | 7<br>6<br>5         |
| 3<br>Host-to-Host<br>Transportschicht   | Verbindung zweier<br>Netzteilnehmer zur<br>Datenübermittlung | TCP, UDP,<br>TLS (SSL),<br>SCTP                                       | Segment   | 4                   |
| 2<br>Internet<br>Internetschicht  | Paketvermittlung<br>Wegewahl (Routing)                       | IP mit ICMP<br>EIGRP (IGRP)<br>RIP, BGP, OSPF                         | Datagramm | 3                   |
| 1<br>Network Access<br>Netzzugangsschicht   | Techniken zur<br>Datenübertragung<br>in div. Netztypen       | Ethernet mit CSMA/CD<br>Token-Passing, FDDI<br>PPP, ARP, BOOTP (RARP) | Frame     | 2<br>1              |
| ↔ physikalisches Medium, Übertragungspfad, Kupfer- und Glasfaserleitungen, Funkstrecken ↔ |  |   |           |                     |

Abbildung 2.3: Die wichtigsten Protokolle im TCP/IP-Referenzmodell

**Quelle:** <https://www.elektroniktutor.de/internet/tcpip.html>

### 2.1.4 HTTP

Das Hypertext Transfer Protocol (HTTP) ist ein zustandsloses und unidirektionales Datenübertragungsprotokoll in einem Netzwerk. Es wird hauptsächlich eingesetzt, um die Dateien vom Server anzufordern und sie in den Browser zu laden und darzustellen. Bei HTTP handelt es sich um eine unverschlüsselte Kommunikation. Dies hat zur Folge, dass alle Informationen im Klartext gesendet werden. Für die verschlüsselte Verbindung bietet sich das sichere HyperText-Übertragungsprotokoll HTTPS<sup>3</sup> an. HTTP arbeitet nach dem Client-Server-Modell (**Abbildung 2.4**).

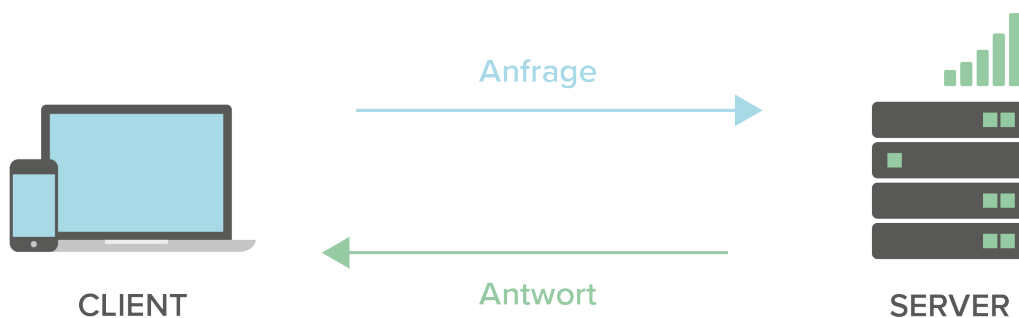


Abbildung 2.4: Das Client-Server-Modell

**Quelle:** <https://www.placetel.de/ratgeber/client>

Der Client (Webbrowser) sendet eine HTTP-Anfrage an den Port 80 des Servers (HTTP-Server). Dieser erledigt die Anfrage vom Client und schickt ihm eine Antwort zurück. Diese Kommunikation verläuft im Textformat. Die Anfrage- sowie die Antwortnachrichten bestehen aus einem Header und Daten. Der Header beinhaltet Steuerinformationen. Der Datenteil enthält den eigentlichen Inhalt der Seite. Nach Abarbeitung der Anfrage wird die Verbindung zwischen Client und Server abgeschlossen. Der Server steht also für die Bearbeitung von neuen Anfragen zur Verfügung. Um dem Server mitzuteilen, was er genau dem Client schicken soll, adressiert der Client bei der Anfrage eine Datei, die sich auf dem Server befindet muss. Dazu verwendet der Client eine URL<sup>4</sup>. Ist diese Datei vom Client nicht vorhanden, antwortet der Server mit der Fehlermeldung (Error 404) zurück. Für eine zuverlässige Kommunikation verwendet HTTP das verbindungsorientierte Transportprotokoll TCP. (vgl. [Stefan Luber, 2018])

<sup>3</sup>Hypertext Transfer Protocol Secure

<sup>4</sup>Uniform Resource Locator



Eine URL ist wie folgt aufgebaut<sup>5</sup>:

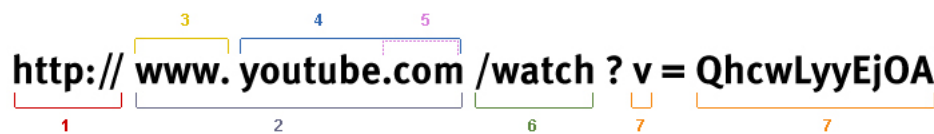


Abbildung 2.5: Aufbau einer URL

**Quelle:** <https://webdesignerinfuehrung.files.wordpress.com/2013/10/url-aufbau.jpg>

1. Das verwendete Protokoll (HTTP). Andere Protokolle könnten ebenfalls verwendet werden, wie HTTPS, FTP.
  2. Es handelt sich um den Host oder Hostnamen.
  3. Die Subdomain: www (World Wide Web).
  4. Die Domain oder der Domainname. Dieser Name ist einmalig, wie eine Postanschrift.
  5. beschreibt die Top-Level-Domain und bezieht sich auf das Ursprungsland der Webseite.
  6. Der Pfad. Dieser verweist auf eine bestimmte Ressource (Datei, Verzeichnis) auf dem Server.
  7. Parameter und Wert: v (Parameter), QhewLyyEjOA (Wert).
- Nach dem Pfad folgt in dem Beispiel ein URL-Parameter. Er wird durch ein Fragezeichen getrennt.

### 2.1.5 Ablauf einer HTTP-Verbindung

Der Ablauf einer HTTP-Verbindung wird mit dem Beispiel eines Aufrufes einer Webseite im Webbrowser dargestellt. Das Aufrufen einer Webseite im Browser erfolgt hauptsächlich in vier Schritten:

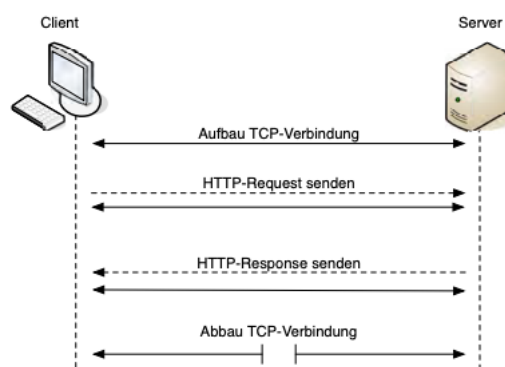


Abbildung 2.6: Klassisches HTTP Request-Response-Paradigma nach [Wöhr, 2004]

<sup>5</sup>vgl. <https://webdesignerinfuehrung.wordpress.com/tag-8/wie-ist-eine-url-aufgebaut/>

1. Der Client baut eine TCP-Verbindung zum Server auf.
2. Der Client, in diesem Fall der Benutzer gibt z.B. eine Adresse (URL) in das Adressfeld seines Webbrowsers ein. Diese Adresse wird als HTTP-Request an der Server gesendet.
3. Der Server bearbeitet die Anfrage von dem Benutzer (Client) und antwortet ihm mit einer HTTP-Response zurück.
4. Nach dem Response baut der Server die Verbindung wieder ab.

### 2.1.6 AJAX

AJAX<sup>6</sup> ermöglicht, dass sich die Daten zwischen Browser und Server im Hintergrund austauschen können, ohne die Seite komplett neu zu laden. Man spricht von einer asynchronen Datenübertragung zwischen Client und Server.

Dabei ist das XMLHttpRequest<sup>7</sup>-Objekt in JavaScript für die Durchführung dieser asynchronen Datenübertragung zwischen Client und Server verantwortlich. XHR ist eine Schnittstelle zwischen JavaScript und Daten auf dem Server. Das XMLHttpRequest sendet eine HTTP-Anfrage an einen Webserver. Die Rückgabe vom Server kann ein JavaScript direkt per DOM<sup>8</sup> und CSS<sup>9</sup> in das Dokument ergänzen oder verändert, ohne die Seite neu laden zu müssen. Die statischen Inhalte bleiben erhalten, während nur veränderliche Information ergänzt werden. Das spart vor allem Zeit, reduziert den Trafficverbrauch und ermöglicht dem Nutzer interaktiv mit dem Server zu kommunizieren.

Nach [o.V., 2017] unterstützt XHR neben XML-Dokumente auch alle Textformate und kann eine Anfrage ebenfalls über HTTPS übermitteln. Ein typisches Beispiel für die AJAX-Anwendung ist die Autovervollständigung von Google. Sobald der Nutzer die Daten im Suchfeld auf der Google Webseite eingibt, wird dabei automatisch die passende Vorschläge geliefert.

---

<sup>6</sup>Asynchronous JavaScript and XML

<sup>7</sup>kurz: XHR

<sup>8</sup>Document Object Modal

<sup>9</sup>Cascading Style Sheets

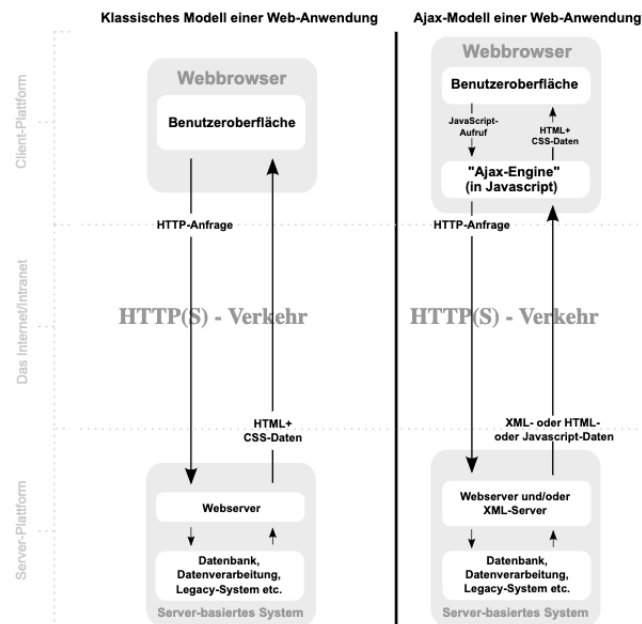


Abbildung 2.7: synchrone und asynchrone Kommunikation

**Quelle:** By I, DanielSHaischt, CC BY-SA 3.0,  
<https://commons.wikimedia.org/w/index.php?curid=2223689>

### 2.1.7 Echtzeit

Der Begriff „Echtzeit“ rückt immer mehr insbesondere bei bestimmten Webanwendungen in den Vordergrund auf. Was genau steckt hinter diesem Begriff? Lutz Schmitt hat in seiner Diplomarbeit diesen Begriff folgendermaßen definiert: „Echtzeit beschreibt die Ausführung eines Prozesses in einem so kurzen Intervall, dass für die menschliche Wahrnehmung keine Zeit vergangen ist. Echtzeit beschreibt also das Phänomen der Reduzierung eines (Maschinen-)Prozesses auf einen Zeitpunkt.“ Ein Grund dafür, dass der Begriff Echtzeit in den letzten Jahren so viel an Bedeutung gewonnen hat, liegt darin, „[...] dass viele Informationsverarbeitungsprozesse, die bis vor wenigen Jahren noch eine wahrnehmbare Dauer in Anspruch nahmen, so stark beschleunigt worden sind, dass sie eben nicht mehr wahrzunehmen sind. [...] Aus der Rechenzeit, die ein Computer für eine bestimmte Aufgabe benötigt, wird die Prozessverarbeitung in Echtzeit, die sofortige Erledigung ohne Verzögerung. [...] Anstatt auf die Maschine warten zu müssen, kann der Mensch unmittelbar weiterarbeiten.“ [Schmitt, 2006]

In der heutigen Zeit, in der das Internet nicht mehr aus unserem Alltag wegzudenken ist, wurde der Begriff Echtzeit in den letzten Jahren so populär, vor allem bei Webanwendungen, die auf eine schnelle und latenzfreie Datenübertragung abhängig sind, wie z.B. Online-Spiele, Chat-Anwendungen oder kollaborative Webseite.

Ein latenzfreier Informationsaustausch zwischen zwei Teilnehmern in einem Netzwerk ist mit dem bekannten Übertragungsprotokoll HTTP nicht gewährleistet. Dieses Protokoll arbeitet, wie bereits bekannt, nach dem Client-Server-Modell (**Abbildung 2.4**). Das hat zu bedeuten, dass nur der Client die Verbindung zum Server aufbaut, nie umgekehrt. Erst dann wenn die Verbindung zum Server erfolgreich hergestellt ist, folgt das Abschicken von Request- und Response-Nachrichten zwischen Client und Server. Man spricht hier von synchroner Übertragung. Nach dem Absenden der Antwortnachricht baut der Server die Verbindung anschließend wieder ab.

Das Übertragungsprotokoll HTTP war in der Vergangenheit die perfekte Lösung für viele klassische Webanwendungen, um Kommunikation oder auch Interaktion zwischen zwei Kommunikationspartnern im Internet zu realisieren. Im Sinne der Echtzeit-Webanwendungen erfüllt dieses Übertragungsprotokoll jedoch nicht alle Anforderungen. Da es sich bei HTTP um eine synchrone Datenübertragung handelt, wird dabei die Benutzeraktivität unterbrochen, bis der Client die Antwort vom Server erhalten hat. Dieser Mangel kann durch AJAX mit der sogenannten asynchronen Datenübertragung behoben werden. Die Daten werden bei diesem Kommunikationsmodell im Hintergrund ausgetauscht, ohne dass die komplette Seite neu geladen werden muss. Trotz der Anwendung von AJAX bleibt das Hauptproblem weiterhin bestehen. Der Server kann bei HTTP nur auf Anfragen eines Clients reagieren, d.h. er wartet passiv auf Anforderungen. Eine Echtzeit-Anwendung soll durch die Interaktion vom Benutzer nicht unterbrochen werden. Häufig werden Echtzeit-Anwendungen durch Hacks (Polling oder Long Polling) simuliert. (vgl. [Herb, oD])

Beim Polling wird der Server in regelmäßigen Abständen (z.B. alle zwei Sekunden) vom Client angefragt, ob er neue Daten hat. Falls neue Daten vorliegen, wird der Server diese ohne Verzögerung an dem Client senden. Im Fall, dass keine Daten für die Anfrage vorliegen, wird dem Client vom Server mit einer leeren Nachricht geantwortet (**Abbildung 2.8**). (vgl. [Aderhold, 2013])

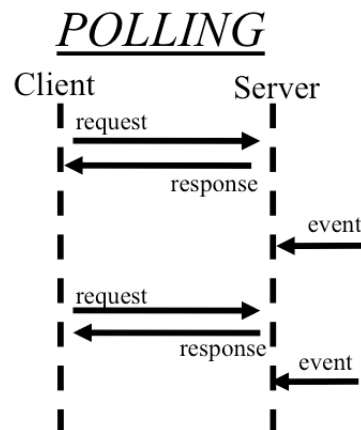


Abbildung 2.8: Polling

**Quelle:** <https://www.heise.de/developer/imgs/06/6/7/6/2/3/3/Polling-61cb54a128001c08.png>

Beim Long Polling wird der Server ebenfalls angefragt. Anders als beim Polling wird der Server diesmal bei nicht vorhandenen Daten solange warten, bis er sie an dem Client liefern kann. Das heißt, der Server hält die Verbindung solange offen, bis neue Daten für den Client verfügbar sind. Nachdem der Client die Daten erhalten hat, sendet er wieder eine Anfrage an den Server, um auf weitere Daten zu warten (**Abbildung 2.9**). (vgl. [Aderhold, 2013])

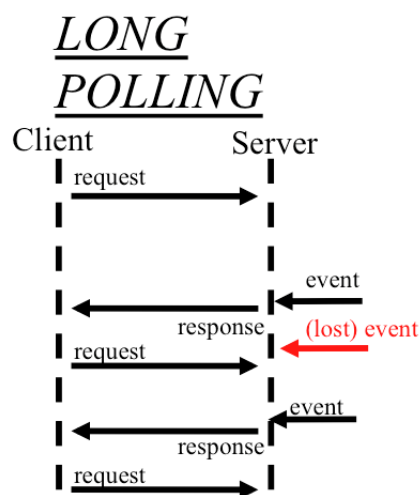


Abbildung 2.9: LongPolling

**Quelle:** <https://www.heise.de/developer/imgs/06/6/7/6/2/3/3/LongPolling-616183343d043825.png>

Damit der Client und Server mit möglichst geringen Latenzen kommunizieren können, wird dafür eine bidirektionale Kommunikation benötigt. Mit dieser Art der Kommunikation können Daten in beide Richtungen gleichzeitig übertragen werden. Man bezeichnet diese Kommunikationsart als Vollduplex. Im Gegensatz zu Vollduplex erlaubt das Halbduplex-Verfahren keine gleichzeitige Kommunikation in beide Richtungen (vgl. [Wikipedia, 2018]).

„HTTP ist von Natur aus ‘nur’ halbduplex. Das bedeutet, dass für die bidirektionale Kommunikation zwischen Browser und Server ein separater HTTP Request für jede Richtung benötigt wird. Das erzeugt natürlich einen Menge Overhead. [...] HTTP Request/Response Header können schnell ein paar Hundert Bytes veranschlagen. Hinzu kommt ab und an die eigentlich wertlose Information, dass es keine Änderungen am Zustand des Servers gab.“ [Weßendorf, 2011]

Um dieses Problem zu lösen, wurde deshalb der Kommunikationsstandard namens WebSocket entwickelt.

### 2.1.8 WebSocket

WebSocket wurde 2008 entwickelt. „Chrome war 2009 der erste Browser, der WebSocket unterstützte; nach und nach folgten alle großen Wettbewerber. Seit 2011 ist WebSocket ein W3C<sup>10</sup>-Standard.“ [o.V., oJ]

WebSocket ist ein bidirektionaler und vollduplexer Kommunikationsstandard, der entwickelt wurde, „[...] um eine bidirektionale Verbindung zwischen einer Webanwendung und einem WebSocket-Server bzw. einem Webserver, der auch WebSockets unterstützt, herzustellen.“ [Wikipedia, 019a]

Mit WebSocket werden Daten in beide Richtungen über einen Kommunikationskanal übertragen. Client und Server können gleichzeitig miteinander „reden“, sobald eine WebSocket-Verbindung besteht. WebSocket verwendet den gleichen Port wie HTTP, nämlich den Port 80. Es wird dabei ein WebSocket-Protokoll namens **Handshake** benutzt, um die Verbindung zwischen Client und Server aufzubauen. (vgl. [Weßendorf, 2011])

---

<sup>10</sup>World Wide Web Consortium

Das Handshake-Verfahren funktioniert wie folgt:

**Client:**

```
GET /chatService HTTP/1.1
Host: server.example.com
Upgrade: websocket
Connection: Upgrade
Sec-WebSocket-Key: dGhlIHNhbXBsZSBub25jZQ==
Sec-WebSocket-Origin: http://example.com
Sec-WebSocket-Protocol: chat, superchat
Sec-WebSocket-Version: 8
```

Abbildung 2.10: Beispiel einer Client-Handshake-Anfrage

**Quelle:** <https://www.heise.de/developer/artikel/WebSocket-Annaeherung-an-Echtzeit-im-Web-1260189.html?seite=all>

Im Prinzip ist es ein Aufsatz, der praktisch auf dem HTTP-Protokoll läuft. Wie in der **Abbildung 2.10** zu sehen ist, schickt der Client eine normale GET-Anfrage an den Server und sagt dementsprechend auf der Serverseite, was er genau haben will. Mit dem **Upgrade** sagt der Client, dass er auf das WebSocket-Protokoll wechseln möchte. Dafür wird für den Verbindungsaufbau einen **Sec-WebSocket-Key** zum Server übermittelt. Bei diesem Key handelt es sich um eine Base64-encodierte Zeichenkette, welche vom Server benutzt wird, um den Verbindungsaufbau zu akzeptieren. (vgl. [Weßendorf, 2011])

**WebSocket-Server:**

```
HTTP/1.1 101 Switching Protocols
Upgrade: websocket
Connection: Upgrade
Sec-WebSocket-Accept: s3pPLMBiTxaQ9kYGzzhZRbK+xOo=
Sec-WebSocket-Protocol: superchat
```

Abbildung 2.11: WebSocket-Server Handshake

**Quelle:** <https://www.heise.de/developer/artikel/WebSocket-Annaeherung-an-Echtzeit-im-Web-1260189.html?seite=all>

Der WebSocket-Server (**Abbildung 2.11**) bearbeitet die Anfrage und antwortet mit dem HTTP-Status code 101 Switching Protocols. Dabei liefert er dem Client die Informationen mit, dass er das Upgrade akzeptiert hat (**Sec-WebSocket-Accept**). (vgl. [Weßendorf, 2011])

„Zusätzlich gibt der Server an, dass er das ‘superchat’-Protokoll kennt. Das hat den Vorteil, dass die Browseranwendung direkt gegen dieses Protokoll beziehungsweise diese API geschrieben wird, statt gegen die WebSocket-API. Entwickler, die mit der Programmierschnittstelle beziehungsweise dem anwendungsspezifischen Protokoll vertraut sind, brauchen keine neue API erlernen, um WebSocket-Anwendungen zu erstellen. Die clientseitige Schnittstelle des ‘superchat’-Protokolls kapselt die eigentliche Kommunikation mit dem WebSocket-Server.“ [Weßendorf, 2011]

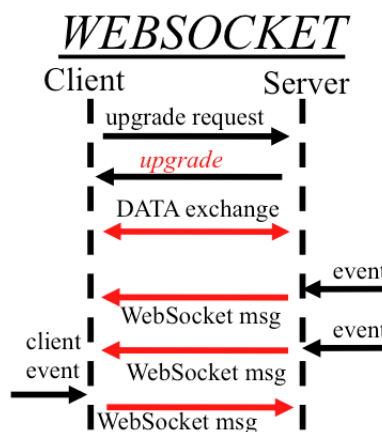


Abbildung 2.12: Das WebSocket-Handshake

**Quelle:** <https://www.heise.de/developer/imgs/06/6/7/6/2/3/3/WebSocket-a70195c3f57b1308.png>

Nach dem Handshake besteht eine persistente Verbindung zwischen Client und Server und beide können jederzeit mit dem Senden von Daten beginnen (**Abbildung 2.12**). Eine WebSocket-Verbindung erkennt man an das neue URL-Schema. Statt wie gewohnt das „http:“ oder für sichere HTTP-Verbindungen das „https:“ als Protokoll anzugeben, wird bei einer WebSocket-Verbindung das „ws:“ verwenden. Für sichere Verbindungen steht das „wss:“ zur Verfügung.

Zusammenfassend lässt sich sagen, dass WebSocket ein geeigneter Protokoll ist, um

„[...] die komplexen Probleme des ‘Echtzeit-Web’, wie Latenz oder Netzverkehr, anzugehen. WebSocket wird jedoch nicht als ein ‘besseres AJAX’ entwickelt. Ebenfalls stellt WebSocket keinen 1:1-Ersatz für HTTP dar, sondern bietet vielmehr einen effizienten, bidirektionalen Kommunikationskanal an. Die Integration von ‘Echtzeit’ innerhalb von Webanwendungen ist nicht mehr an Hacks und Workarounds gebunden, sondern erfolgt auf Basis eines standardisierten, effizienten und bidirektionalen Protokolls. Wichtig ist hierbei, dass man sämtliche TCP/UDP-Protokolle auf Basis von WebSocket zum Browser bringen kann. Der Abstraktionsgrad zu-



künftiger Webanwendungen steht damit den Desktop-Anwendungen in nichts nach.“  
[Weßendorf, 2011]

### 2.1.9 jQuery

jQuery ist eine JavaScript-Bibliothek, die Klassen und Methoden zur Verfügung stellt, um die Arbeit mit JavaScript zu vereinfachen. jQuery ist nicht nur kompakter und komfortabler als JavaScript, sondern außerdem browserübergreifend, was bei JavaScript in der Vergangenheit nicht immer der Fall war. jQuery vereinfacht viele JavaScript-Funktionen, die bei der Webentwicklung oft verwendet wird. Wie JavaScript ermöglicht jQuery auch den Zugriff auf DOM-Elemente. DOM-Elemente können gezielt angesprochen und manipuliert werden. (vgl. [Augsten, 2019])

Laut [Wikipedia, 019b] ist jQuery die meist verwendete JavaScript-Bibliothek und wird auf rund 70% der 10000 meistbesuchten Webseiten eingesetzt.

Das DOM versteht sich als die Schnittstelle für den Zugriff auf den Tags, Attribute und Inhalte von HTML- oder XML-Dokumenten und wird vom W3C definiert. Mithilfe von Selektoren und dem DOM können die HTML-Elemente aufgerufen, verändert, hinzugefügt und gelöscht werden. Die Elemente können über id- oder class-Attribute selektiert werden.

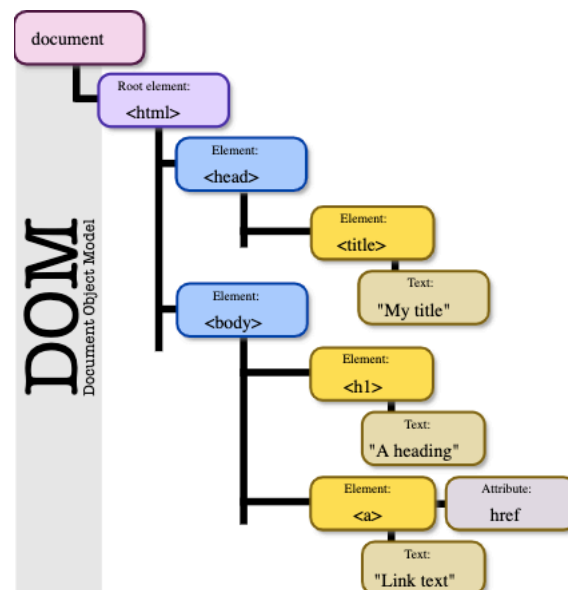


Abbildung 2.13: DOM - Elementenbaum einer Webseite

**Quelle:** By Birger Eriksson - Own work, CC BY-SA 3.0,  
<https://commons.wikimedia.org/w/index.php?curid=18034500>

Weitere jQuery-Funktion:

- Event Handling
- Form Handling
- AJAX
- JSON
- Collect und Select
- Animationen

## 2.2 Responsive Webdesign

Um eine Webseite geräteübergreifend zu gestalten, benötigt sie ein „Responsive Webdesign“. Beim Responsive Webdesign handelt es sich um eine reaktions- und anpassungsfähige Weboberfläche, so dass diese ein einheitliches Anzeigen von Inhalten sowie den strukturellen Aufbau einer Webseite auf dem Desktop-Computer, Tablet und Smartphone bietet. Laut [Wikipedia, 019c] wurde der Begriff im Jahr 2010 vom amerikanischen Webdesigner Ethan Marcotte erfunden.

Da es zu jener Zeit kaum internetfähige Mobilgeräte auf dem Markt gab, waren die meisten Webseiten statisch und nur für Desktop-Computer entwickelt worden. Um eine Webseite für Mobilgeräte anzubieten, musste separat eine mobile Webseite entwickelt werden.

Erst mit der Einführung des iPhones von Apple im Januar 2007<sup>11</sup> war es für den Benutzer möglich, mobil ins Internet zu gehen. Erstmals verfügte ein Mobilgerät einen vollwertigen Webbrowser und war über einen Touchscreen verfügbar. Viele Firmen folgten dem Beispiel von Apple und wenige Jahre später brachten sie eine Menge an verschiedenen Mobilgeräten mit individuellen Displaygrößen auf dem Markt.

Seit der Einführung der Smartphones, Tablets und mit der steigenden mobilen Internetnutzung ist ein Responsive Webdesign heutzutage nicht nur ein nettes Feature sondern ein Pflichtprogramm für jeden Webseitenbetreiber. Für neue Webseiten liegt es auf der Hand, wie man vorgehen sollte, für bestehende statische Webseiten ist der Aufwand sehr groß, diese auf eine responsive Webseite umzustellen, da eventuell die komplette Seite neu entwickelt werden muss.

Die vorliegende Arbeit soll auf verschiedenen Geräten optimal angezeigt werden können. Als „responsive“ Webseite muss sie unter anderem folgende Eigenschaften haben.

---

<sup>11</sup>vgl. [https://de.wikipedia.org/wiki/IPhone\\_\(erste\\_Generation\)](https://de.wikipedia.org/wiki/IPhone_(erste_Generation))

- **flexibles Grid<sup>12</sup>-Layout**

Seiten und Elemente, wie Bilder oder Textblöcke, müssen sich der Bildschirmauflösung des mobilen Endgeräts anpassen. Hier werden für die Elemente und Seiten prozentuale statt fester Pixelwerte verwendet.

- **keine festen Schriftgrößen**

Fließtexte und Headlines müssen so angepasst werden, dass sie sowohl auf dem PC als auch auf Smartphones und Tablet gut lesbar sind. Hier ist eine feste Schriftgröße nicht geeignet. Es wird daher mit prozentualen Werten oder Maßen wie em gearbeitet.

## 2.3 Web 2.0

Das Web 2.0 wird als „Mitmach-Netz“ verstanden. Es ist eine Revolution hinsichtlich der Nutzung des World Wide Web, bei der Internetnutzer nicht mehr wie früher, nur die Inhalte des Internets konsumieren, sondern auch eigene Inhalte selbst produzieren, wie Videos und Fotos einstellen oder Texte schreiben. Durch seine Beteiligung im Web ist der Internetnutzer selbst ein Teil des Internets.

Beim Web 2.0 handelt es sich dabei nicht um eine neue Technologie, sondern vielmehr um die Weiterentwicklung des Internets. Dazu zählt beispielsweise der vermehrte Einsatz der AJAX-Technologie, die eine asynchrone Datenübertragung zwischen Client und Server ermöglicht. Oder der Einsatz von Abonnementdienste (Web-Feeds), welche eine grundlegende Eigenschaft für die neue Generation des World Wide Webs sind. (vgl. [o.V., 2008])

Der Begriff Web 2.0 ist durch seinen Artikel „What is Web 2.0“ von amerikanischen Verleger Tim O'Reilly in 2005 bekannt geworden<sup>13</sup>.

---

<sup>12</sup>aus dem Englischen für das Gestaltungsraster bekannt

<sup>13</sup>vgl. <https://www.oreilly.com/pub/a/web2/archive/what-is-web-20.html>

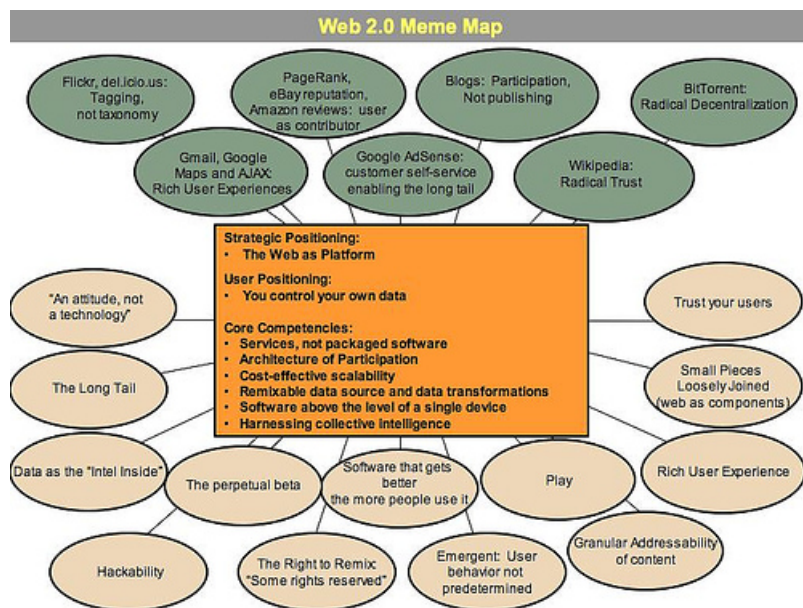


Abbildung 2.14: Das Konzept Web 2.0 nach einer Brainstorming-Sitzung

**Quelle:** <http://www.siliconbeat.com/entries/meme-map.jpg>

Die **Abbildung 2.14** zeigt ein Konzept von Web 2.0, die mit einem Brainstorming zwischen O'Reilly und MediaLive International entwickelt wurde. Die **Abbildung 2.15** beschreibt einige Beispiele für die Bedeutung von Web 2.0, die im ersten Brainstorming zwischen O'Reilly und MediaLive International formuliert wurden.

| Web 1.0                    |   | Web 2.0                    |
|----------------------------|---|----------------------------|
| DoubleClick                | → | Google AdSense             |
| Ofoto                      | → | Flickr                     |
| Akamai                     | → | BitTorrent                 |
| mp3.com                    | → | Napster                    |
| Britannica Online          | → | Wikipedia                  |
| personal websites          | → | blogging                   |
| evite                      | → | upcoming.org and EVDB      |
| domain name speculation    | → | search engine optimization |
| page views                 | → | cost per click             |
| screen scraping            | → | web services               |
| publishing                 | → | participation              |
| content management systems | → | wikis                      |
| directories (taxonomy)     | → | tagging ("folksonomy")     |
| stickiness                 | → | syndication                |

Abbildung 2.15: Die Bedeutung von Web 2.0

**Quelle:** <https://www.oreilly.com/pub/a/web2/archive/what-is-web-20.html?page=1#mememap>

Außerdem stellte O'Reilly sieben Merkmale vor, die kennzeichnend für Web 2.0 sind:

- **Das Web als Plattform:**
  - Das Web als Plattform ähnlich wie ein Betriebssystem.
- **Kollektive Intelligenz:**
  - Verlinkung der Daten und Seiten untereinander.
- **Daten als nächstes Intel Inside:**
  - Die gesammelten Daten sind die Basis einer Webanwendung und sind wichtiger und wertvoller als eine einzelne Anwendung.
- **Softwarelebenszyklus:**
  - Software wird nicht mehr als Produkt ausgeliefert, sondern als Service.
- **Lightweight Programming Models:**
  - Daten werden durch Web-Services bereitgestellt.
  - Die Daten werden über den Web-Services wie RSS oder REST-basierten Web-Services verteilt oder ausgetauscht.
- **Software über Gerätegrenzen hinaus:**
  - Geräteunabhängige Anwendungen, z.B. nicht nur für den PC sondern auch mobile Geräte.
- **Rich User Experiences:**
  - Benutzerführung mit interaktiver Benutzeroberfläche, die sich kaum von einem Desktop-Programm unterscheiden.
  - AJAX-Technologie.

## 2.4 Rich Internet Applications

Die klassischen Webanwendungen schicken generell die Formulareingabe vom Nutzer an einen Webserver. Als Rückgabe vom Webserver bekommt der Client, in diesem Fall der Webbrowser, eine neu generierte Webseite gemäß dem zuvor eingegebenen Daten. Bei jeder Anfrage des Clients wird die komplette Webseite neu generiert und übertragen. Demzufolge werden solche Webanwendungen als wenig intuitiv bezeichnet und kann auch nachteilig auf die Geschwindigkeit und schlechten User-Experience führen.

Außerdem bieten klassische Webanwendungen nicht genügend Funktionalitäten und Interaktionsmöglichkeiten auf der Benutzeroberfläche. Das hat zur Folge, dass sie gegenüber den Desktop-Anwendungen im Nachteil sind.

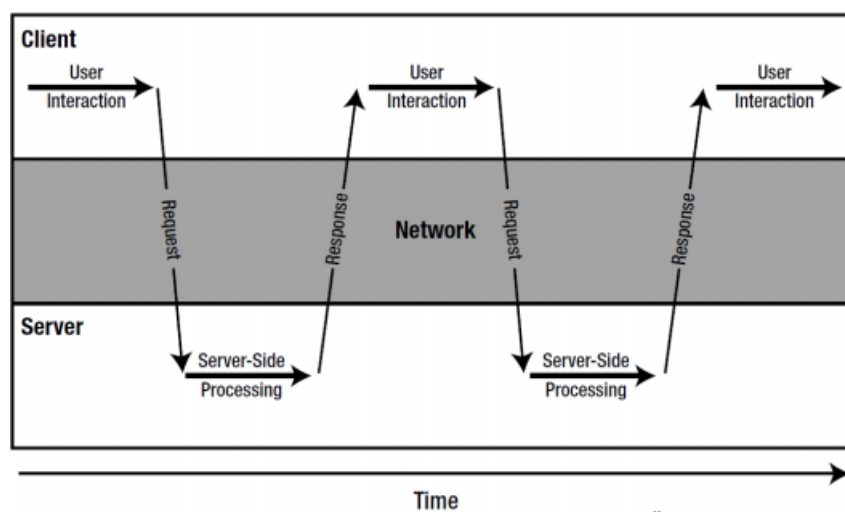


Abbildung 2.16: Kommunikationsmodell einer klassischen Webanwendung

Quelle: [Bankras/Smeets/Bram/Boness/Uri/Roald, 008a]

RIAs (Rich Internet Applications) sind webbasierte Anwendungen, deren Funktionalitäten und Eigenschaften denen einer Desktop-Anwendungen entsprechen. RIAs erkennt man unter anderem daran, dass sie plattformunabhängig sowie über das Internet zugänglich sind, wobei RIAs auch offline Anwendungen bieten. Und sie sollten auch eine interaktive Benutzeroberfläche erzeugen können. Zu Interaktionsmöglichkeiten auf der Benutzeroberfläche zählen beispielsweise Drag-and-Drop-Fähigkeit oder Bedienbarkeit über Tastenkürzel.

Nach [Hartmann, 2007a] unterscheidet sich die RIA-Technologie in zwei Kategorien:

- **rein Browser-basierte Anwendungen:** Der Client verwendet ausschließlich einen Browser. In dieser Kategorie laufen alle AJAX-Anwendungen ohne weitere Browser-Plugins.
- **in plugin-basierte und Stand-Alone-Anwendungen:** Auf der Clientseite werden weitere zusätzliche Laufzeitumgebungen (Flash oder Java) verwendet.

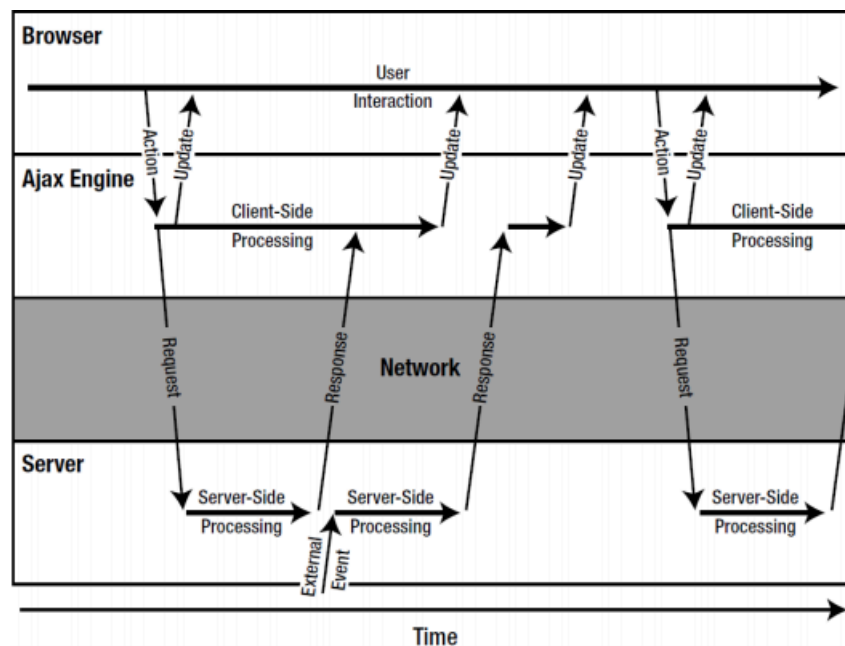


Abbildung 2.17: Kommunikationsmodell einer RIA am Beispiel einer AJAX-Anwendung

**Quelle:** [Bankras/Smeets/Bram/Boness/Uri/Roald, 008b]

Die Vorteile der RIA gegenüber herkömmlichen Webanwendungen können nach [Hartmann, 2007b] wie folgt beschrieben werden:

- **flüssigere Bedienung** im Vergleich zur herkömmlichen Webanwendungen, da nur notwendige Teile der GUI verändert werden.
- geringer Serverzugriff, da in RIAs **mehr Daten und Logik auf Clientseite** verarbeitet werden.
- geringer Serverzugriff und eine flüssigere Bedienung führen zu **weniger Netzwerklast**.
- **bekannte Benutzerschnittstelle**, da sie sich an Desktop-Anwendungen orientiert, ist für den Benutzer keine große Umgewöhnung.
- **einfacher Zugriff auf RIA**, da die Laufzeitumgebung, insbesondere der Browser in den meisten Fällen auf den Clients bereits installiert ist.
- **offline-Funktionalität**, wenn die Programmlogik der konzipierte Anwendung auf Clientseite abläuft, kann das Programm auch komplett ohne Serverzugriff verwendet werden, sobald es einmal geladen ist.

| Feature                        | C/S, Desktop | Web     | RIA    |
|--------------------------------|--------------|---------|--------|
| Universal client (browser)     | YES          | YES     | YES    |
| Client installation            | Complex      | Simple  | Simple |
| Interaction capabilities       | Rich         | Limited | Rich   |
| Server-side business logic     | YES          | YES     | YES    |
| Client-side business logic     | YES          | Limited | YES    |
| Full page refresh required     | NO           | YES     | NO     |
| Frequent server round-trips    | NO           | YES     | NO     |
| Server-to-client communication | YES          | NO      | YES    |
| Disconnected functioning       | YES          | NO      | YES    |

Abbildung 2.18: Vergleich der Desktopanwendung, klassischen Webanwendung und RIAs

**Quelle:** [Bozzon/Comai/Fraternali/Carughi, 2006]

## 2.5 Thick Client und Thin Client

Ein Client ist ein Computer oder eine Software, die eine Netzwerkverbindung zu einem Server aufbaut, mit ihm kommuniziert und Dienste sowie Daten anfordert. Ein Client hat vor allem die Aufgabe, Anfragen der Anwender an den Server zu übermitteln und die übermittelte Informationen grafisch darzustellen, sodass der Anwender sie auf seinem Gerät wahrnehmen kann (**siehe Abbildung 2.4**).

Es gibt dabei zwei Arten von Clients zu unterscheiden:

- **Thick Client:**  
wird häufig auch als Fat-Client genannt, ist ein leistungsfähiger Desktop-Computer, der die Daten lokal selbst verarbeitet.
- **Thin Client:**  
ist in Abhängigkeit mit einem Server, um die Aufgaben verarbeiten zu können. Thin Client hat nur eine Rolle, nämlich Anfragen an der Server weiterzuleiten und die vom Server übermittelten Daten auszuwerten. Kein Installationsaufwand, da er über einen Webbrowser läuft.



## 3 Analyse

Dieses Kapitel der Arbeit betrachtet zunächst den gegenwärtigen Stand der Technik hinsichtlich bereits vorhandenen Tools. Ihre Stärken und Schwächen werden ebenfalls aufgezeigt. Danach geht es um die Anforderungsanalyse der Webanwendung. Dazu wird eine allgemeine Struktur festgelegt, wie die Arbeit systematisch aufgebaut sein soll. Anschließend wird der aktuelle Zustand (Ist-Analyse) des Projektes ermittelt und anhand dieser Ist-Analyse erfolgen die funktionalen und nicht-funktionalen Anforderungen an die zu entwickelnde Webanwendung.

### 3.1 Stand der Technik

Bei der Suche nach öffentlich zugänglichen Tools für die Durchführung von Workshops wurden folgenden Ergebnisse gefunden:

#### 3.1.1 IdeaBoardz

IdeaBoardz<sup>1</sup> ist eine freie webbasierte Anwendung zum Brainstorming, Erstellen einer ToDo-Liste oder zur Retrospektive im agilen Projektmanagement. Mit diesem Tool ist eine Zusammenarbeit möglich. Die beteiligten Personen können entweder zeitgleich oder zu verschiedenen Zeiten ortsunabhängig auf das gemeinsame Dokument zugreifen und bearbeiten. In Echtzeit zusammenarbeiten, ist bei IdeaBoardz nicht realisierbar. Das hat zur Folge, dass bei der Veränderung des Zustands keine sofortige Aktualisierung der Benutzeroberfläche erfolgt. IdeaBoardz wird unter anderem bei Brainstorming-Methode wie die 6-Hüte-Methode<sup>2</sup> von De Bono und auch für die Ideenbewertung bekannte SWOT<sup>3</sup>-Analyse angewendet. Die Registrierung ist optional, so dass der Nutzer auch IdeaBoardz verwenden kann, ohne sich anzumelden.

---

<sup>1</sup>vgl. <https://ideaboardz.com/>

<sup>2</sup>engl. Six Thinking Hats

<sup>3</sup>steht für Strengths-Weaknesses-Opportunities-Threats-Analyse

**IdeaBoardz** brainstorm, retrospect, collaborate

**How**

- [Create](#) my own IdeaBoard
- [Share](#) the url

Name \*

Description \*

Format  
2 Sections

Section Title \*

☐ Ich bin kein Roboter. reCAPTCHA Datenschutzerklärung - Nutzungsbedingungen

Create

Test Drive...  
Lets take it for a [Spin!](#)

Feedback  
• [via Twitter](#)  
• [via Board](#)

483931 Boardz

Best viewed on  
[chrome](#), [firefox](#) or [safari](#)

Abbildung 3.1: Erstellen eines eigenen IdeaBoards.

Die **Abbildung 3.1** zeigt, wie ein IdeaBoard zu erstellen ist. Neben dem Namen des Boards werden das Thema (Description) und Formate (Format) benötigt. Es können bis zu 10 Sektionen gewählt werden und es stehen noch weitere Formate zur Verfügung, wie Pro und Contra, ToDo-Liste, Six Thinking Hats und vieles mehr. Anschließend wird ein Titel für jegliche Sektionen eingegeben.

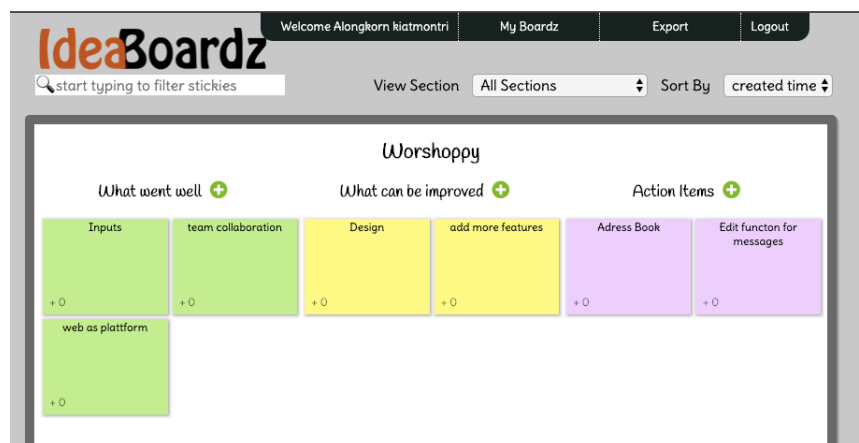


Abbildung 3.2: Darstellung von Sektionen

Wie in **Abbildung 3.2** zu sehen ist, sind die Eingaben in Sektionen strukturiert und farbig sortiert. Das Thema steht in der Mitte. Jede Sektion hat einen Titel und einen Plus-Button. Mit diesem Button können zu jeder Sektion neue Eingaben hinzugefügt werden. Die Eingaben werden als Karteikarten bzw. Notizzetteln visualisiert. Der weiße Hintergrund kann wie ein Whiteboard oder eine Pinnwand gesehen werden. Die Eingaben können auch von beteiligten Personen abgestimmt werden. Es ist auch möglich,

die Daten nach Datum oder Abstimmung sortieren zu lassen.

Als weiteres Feature lassen sich die Sektionen einzeln darstellen (**Abbildung 3.3**). Die Suche nach dem Eingabeinhalt und das Exportieren der Ergebnisse sowohl als PDF-Datei als auch in ein Excel-Dokument werden ebenfalls bei dieser Webanwendung angeboten. Die Daten können sowohl innerhalb als auch außerhalb der Sektion zusammengeführt (Merge) werden. Ebenso lassen sich die Daten aus einer Sektion anderer Daten per Drag & Drop zuordnen, wie in **Abbildung 3.4** zu sehen ist. Mit dem Teilen von URL kann das jeweilige IdeaBoardz für die Zusammenarbeit freigegeben werden.

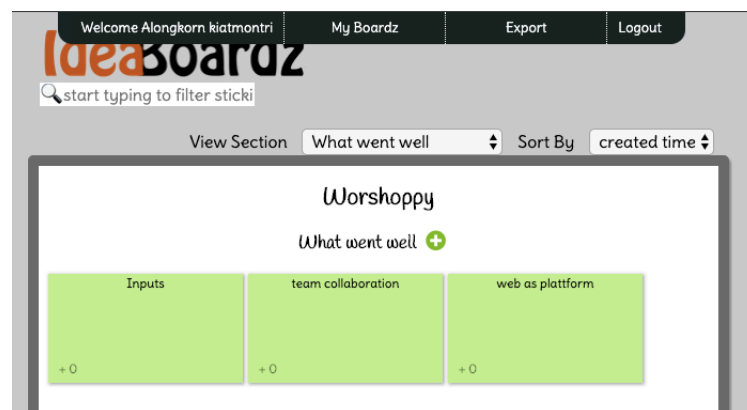


Abbildung 3.3: Darstellung einer der Sektionen

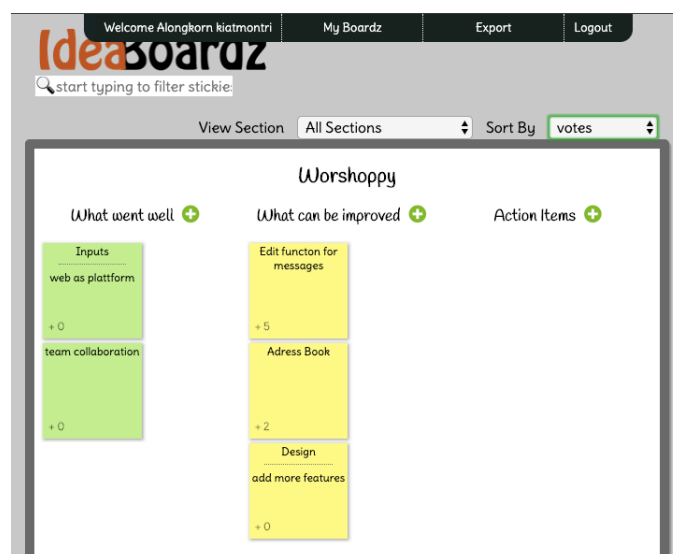


Abbildung 3.4: Zusammenführen und Zuordnen von Daten

Einige der oben dargestellten Features können für das Workshopy-Projekt übernommen werden. Zu nennen sind:

- Die Eingabe wie ein Notizzettel oder Karteikarten visualisieren.
- Die Ergebnisse als PDF-Datei exportieren.
- Eingaben in Sektion darstellen.
- Zuordnung von Daten (per Drag & Drop).

Mit welchen Webtechnologien IdeaBoardz entwickelt wurde, lässt sich anhand der Informationen auf der Webseite nicht erkennen. Man kann aber davon ausgehen, dass es sich bei IdeaBoardz um eine webbasierte Anwendung mit reichlich Interaktionen auf der Benutzeroberfläche handelt, d.h. es ist über einen Webbrowser nutzbar und der Nutzer muss nichts installieren. Dementsprechend gehört IdeaBoardz zu einer Thin Client-Anwendung und zählt auch zu Rich Internet Applications sowie Web 2.0-Anwendung. (siehe Kapitel 2).

### 3.1.2 Miro-RealtimeBoard

Miro<sup>4</sup> ist eine dynamische Webanwendung und handelt es sich um ein kollaborativer Online-Whiteboard in Echtzeit. Um das Online-Whiteboard nutzen zu können, wird ein Account benötigt. Dafür muss man sich bei Miro registrieren. Miro bietet die kostenlose Version an, sie ist für bis zu drei Teammitglieder und drei Boards geeignet.

Begonnen wird mit einer leere Seite oder man verwendet eine von Miro bereitgestellten Vorlage. Zur Vorlage gehören unter anderem MindMap, Flowchart, Brainwriting und Concept Map. Einfügen neuer Dateien, Bilder und Dokumenten aus Google Drive oder vom Rechner ist auch möglich, um Informationen auszutauschen. Der Nutzer kann virtuelle Notizen erstellen. Die Notizen können sich nach Farbe unterscheiden und per Drag & Drop über das komplette Board verschieben. Mit Hilfe von Share-Button vereinfacht Miro die Teilen-Funktion über eine URL oder einen Gmail-Account das ortsunabhängige und kollaborative Arbeiten in Echtzeit. Somit können die beteiligten Person beispielsweise während des Brainstormings auf die Ideen der anderen eingehen und kommentieren. Außerdem können die Benutzer das Whiteboard in eine Präsentation umwandeln oder als eine PDF-Datei exportieren.

Miro ist ebenfalls gut geeignet zur Umsetzung eines klassischen Brainstormings (**Abbildung 3.5**). Die Ideen werden in Form von Notizen erstellt. Zusammenfassend können die Notizen nach Farben kategorisiert werden.

---

<sup>4</sup>vgl. <https://miro.com/>



Abbildung 3.5: Realisieren eines Brainstormings mit Hilfe von Miro

Folgende Erkenntnisse wurden bei der Analyse von Miro gefunden und werden für das Workshopy-Projekt übernommen:

- AJAX-Anwendung
- Thin Client-Anwendung
- Web 2.0-Anwendung
- Rich Internet Applications

### 3.1.3 MindMap

## **4 Design**

## **5 Implementierung**

## **6 Fazit und Ausblick**



# Literaturverzeichnis

- [Aderhold, 2013] Aderhold, M. (2013). "Skalierbares HTTP Long Polling". In: accso.de, URL: <https://accso.de/magazin/skalierbares-http-long-polling/> (abgerufen am 02.08.2019).
- [Augsten, 2019] Augsten, S. (2019). "Was ist jQuery?". In: Webseite dev-insider, URL: <https://www.dev-insider.de/was-ist-jquery-a-782237/> (abgerufen am: 16.07.2019).
- [Balzert, 2010] Balzert, H. (2010). *Lehrbuch der Softwaretechnik: Basiskonzepte und Requirements Engineering*. 3. Auflage, Spektrum Akademischer Verlag.
- [Bankras/Smeets/Bram/Boness/Uri/Roald, 008a] Bankras/Smeets/Bram/Boness/Uri/Roald (2008a). *"Beginning Google Web Toolkit from Novice to Professional"*. Apress Verlag, ISBN 978-1-4302-1032-0, S. 6.
- [Bankras/Smeets/Bram/Boness/Uri/Roald, 008b] Bankras/Smeets/Bram/Boness/Uri/Roald (2008b). *"Beginning Google Web Toolkit from Novice to Professional"*. Apress Verlag, ISBN 978-1-4302-1032-0, S. 7.
- [Bozzon/Comai/Fraternali/Carughi, 2006] Bozzon/Comai/Fraternali/Carughi (2006). *"Conceptual modeling and code generation for rich internet applications"*. Proceedings of the 6th international conference on Web engineering, ISBN:1-59593-352-2, S.353-360.
- [Chornaya, 2018] Chornaya, J. (2018). "5 effektive Workshop Methoden die außerdem Spaß machen". In: eventbrite.de, URL: <https://www.eventbrite.de/blog/effektive-workshop-methoden/#workshopmethodefishbowl> (abgerufen am 18.07.2019).
- [Hartmann, 2007a] Hartmann, L. (2007a). "Rich Internet Applications". Hochschule für Angewandte Wissenschaften Hamburg, Master Seminararbeit, URL: <https://users.informatik.haw-hamburg.de/~ubicomp/projekte/master07-08/hartmann/bericht.pdf>, S.3, (abgerufen am 22.07.2019).
- [Hartmann, 2007b] Hartmann, L. (2007b). "Rich Internet Applications". Hochschule für Angewandte Wissenschaften Hamburg, Master Seminararbeit, URL: <https://users.informatik.haw-hamburg.de/~ubicomp/projekte/master07-08/hartmann/bericht.pdf>, S.4 f., (abgerufen am 22.07.2019).
- [Herb, oD] Herb, M. (o.D.). "Data Synchronisation zwischen mehreren Clients in Echtzeit". In: doag.org, URL: [https://www.doag.org/formes/pubfiles/2262550/91-2010-K-JAVA-Herb-Data\\_Synchronisation.pdf](https://www.doag.org/formes/pubfiles/2262550/91-2010-K-JAVA-Herb-Data_Synchronisation.pdf) (abgerufen am 2.08.2019).

- [Holtkamp, 2001] Holtkamp, H. (2001). "TCP/IP im Detail". In: rvs.uni-bielefeld.de, URL: [http://www.rvs.uni-bielefeld.de/~heiko/tcpip/tcpip\\_html\\_alt/kap\\_2\\_4.html](http://www.rvs.uni-bielefeld.de/~heiko/tcpip/tcpip_html_alt/kap_2_4.html) (abgerufen am 30.07.2019).
- [Holzer, 2012] Holzer, P. (2012). *Diplomarbeit: Kreativitätstechniken zur Ideenfindung*. Bachelor + Master Publishing, ISBN: 978-3-95549-501-5, S.11.
- [Karadeniz, oJ] Karadeniz, B. (o.J.). "TCP/IP - Haussprache des Internet". In: netplanet.org, URL: <https://www.netplanet.org/aufbau/tcpip.shtml> (abgerufen am 31.07.2019).
- [o.V., 2008] o.V. (2008). "Im Blickpunkt Web 2.0". In: imblickpunkt.grimme-institut.de, URL: <https://imblickpunkt.grimme-institut.de/wp/wp-content/uploads/2014/12/IB-Web20.pdf> (abgerufen am 21.07.2019).
- [o.V., 2017] o.V. (2017). "Javascript XMLHttpRequest (XHR) und Ajax". In: mediaevent.de, URL: <https://www.mediaevent.de/javascript/XmlHttpRequest.html> (abgerufen am 20.07.2019).
- [o.V., 2019] o.V. (2019). "TCP - Transmission Control Protocol". In: elektronik-kompodium.de, URL: <https://www.elektronik-kompodium.de/sites/net/0812271.htm> (abgerufen am 30.07.2019).
- [o.V., oJ] o.V. (o.J.). "WebSocket: So funktioniert die Verbindung zwischen einer Webanwendung und einem Webserver". In: it-times.de, URL: <https://www.it-times.de/news/websocket-so-funktioniert-die-verbinding-zwischen-einer-webanwendung-und-einem-webserver-118921/> (abgerufen am 02.08.2019).
- [phil. Tomas Bohinc, 2016] phil. Tomas Bohinc, D. (2016). "Workshop". In: projektmagazin.de, URL: <https://www.projektmagazin.de/methoden/workshop> (abgerufen am 17.07.2019).
- [Reich, 2007] Reich, P. D. K. (2007). "Brainstorming". In: methodenpool.uni-köln.de, URL: <http://methodenpool.uni-koeln.de/download/brainstorming.pdf> (abgerufen am 18.07.2019).
- [Schmitt, 2006] Schmitt, L. (2006). "Die Wahrnehmung von Zeit und der Einfluss von Design". Köln International School of Design, S.55, URL: [https://lutzschmitt.com/content/3-publications/lutz\\_schmitt-wahrnehmung\\_zeit\\_einfluss\\_design.pdf](https://lutzschmitt.com/content/3-publications/lutz_schmitt-wahrnehmung_zeit_einfluss_design.pdf) (abgerufen am 01.08.2019).
- [Stefan Luber, 2018] Stefan Luber, A. D. (2018). "Was ist HTTP (Hypertext Transfer Protocol)?". In: ip-insider.de, URL: <https://www.ip-insider.de/was-ist-http-hypertext-transfer-protocol-a-691181/> (abgerufen am: 30.07.2019).
- [van Aerssen, oJ] van Aerssen, B. (o.J.). Brainstorming im Florence Innovation Project des verrocchio Institute. In: ideenfindung.de, URL: <https://www.ideenfindung.de/Brainstorming-Kreativitaet%20C3%A4tstechnik-Ideenfindung.html> (abgerufen am 18.07.2018).
- [Weßendorf, 2011] Weßendorf, M. (2011). "WebSocket: Annäherung an Echtzeit im Web". In: heise.de, URL: <https://www.heise.de/developer/artikel/WebSocket-Annäherung-an-Echtzeit-im-Web-1260189.html?seite=all> (abgerufen am 2.08.2019).

- [Wikipedia, 2018] Wikipedia (2018). "Duplex (Nachrichtentechnik)". URL: [https://de.wikipedia.org/wiki/Duplex\\_\(Nachrichtentechnik\)](https://de.wikipedia.org/wiki/Duplex_(Nachrichtentechnik)) (abgerufen am 2.08.2019).
- [Wikipedia, 019a] Wikipedia (2019a). "WebSocket". URL: <https://de.wikipedia.org/wiki/WebSocket> (abgerufen am 02.08.2019).
- [Wikipedia, 019b] Wikipedia (2019b). "jQuery". URL: <https://de.wikipedia.org/wiki/JQuery> (abgerufen am 20.07.2019).
- [Wikipedia, 019c] Wikipedia (2019c). "Responsive Webdesign". URL: [https://de.wikipedia.org/wiki/Responsive\\_Webdesign](https://de.wikipedia.org/wiki/Responsive_Webdesign) (abgerufen am 20.07.2019).
- [Wöhr, 2004] Wöhr, H. (2004). "*Web-Technologie*". 1.Auflage, dpunkt.verlag, ISBN: 978-3898642477, S.219.