

Projektportfolio

Erfassung der Arbeitszeit in der BBQ GmbH

Digitale Zeiterfassungs-Software unter Berücksichtigung des Arbeitszeitgesetzes

Betreuerin	Anna Schmelzle
Bearbeitungszeitraum	24 Wochen
Kurs	TINF-24F, Projektmanagement
Studienrichtung	Informatik - Cybersecurity
Projektteam	Eifrige Otter
Teammitglieder	Fabian Steiß, Luca Schmoll, Magnus Lörcher, Manuel Lutz

Inhaltsverzeichnis

Einleitung	3
1 Vorgehensmodell & Projektstruktur	3
2 Projektorganisation	4
2.1 Liefergegenstände und Meilensteine	5
3 Stakeholder-Analyse.....	7
4 Anforderungsanalyse	8
5 Zeitplan und Budgetberechnung	15
6 Wireframes & Prototyp	16
6.1 Wireframes	17
7 Projektcontrolling	19
8 Qualitätssicherung	20
9 Risikomanagement.....	21
10 Kommunikationskonzept	23
11 Schulungskonzept.....	25
12 Beistellungen BBQ GmbH.....	26
13 Erweiterungspotenzial	27
14 Reflexion & Lessons Learned	27

Einleitung

Die BBQ GmbH ist ein kleines Unternehmen, das sich im Aufschwung befindet. Seit letztem Jahr hat das Unternehmen einen Zuwachs von knapp 300% der Mitarbeiter erhalten. Die BBQ GmbH beschäftigt nun etwa 50 Mitarbeiter. Bevor das Unternehmen diesen starken Zuwachs erlebt hat, hatte jeder Mitarbeiter einen Zettel, auf welchem er seine Arbeits- und Pausenzeiten festgehalten hat. Für ein Unternehmen, mit wenigen Mitarbeitern, hat das auch noch einigermaßen funktioniert. Jetzt jedoch, gibt es jeden Monat 50 Zettel, die durch die Personalabteilung ausgewertet werden müssen. Das ist ein enormer Aufwand an Personenarbeitstagen, der hier verloren geht. Außerdem ist die Personalabteilung nicht mehr in der Lage, zu prüfen, ob die Arbeitszeitgesetze eingehalten werden, oder nicht. Die Geschäftsführung der BBQ GmbH hat dieses Problem erkannt und möchte nun die Initiative ergreifen. Sie haben uns als Dienstleister beauftragt, dieses Projekt in Angriff zu nehmen. Hierzu stellen sie lediglich einige Anforderungen, welche innerhalb des Projekts umgesetzt werden müssen. Ziel dieses Projekts ist es, eine moderne, digitale Zeiterfassungssoftware für die BBQ GmbH zu konzeptionieren und zu entwickeln. Es gilt also nun für uns, das Projektportfolio zu entwerfen und basierend auf diesem, die passende Softwarelösung zu entwickeln.

1 Vorgehensmodell & Projektstruktur

Das Projekt, mit dem die BBQ GmbH uns als Dienstleister beauftragt hat, ist ein komplexes Projekt, bei welchem vielerlei Dinge berücksichtigt werden müssen. Des Weiteren können sich die Anforderungen der BBQ GmbH jederzeit ändern. Sie geben zwar einige Anforderungen bereits vor und lassen uns viel Freiheiten in der Umsetzung, jedoch ist eine Anforderung der BBQ, dass das deutsche Arbeitszeitgesetz eingehalten wird. Dieses Gesetz kann sich aber im längeren Projektzeitraum von 24 Wochen, jederzeit ändern. Da eine Umplanung des Projekts zu so einem Stadium schwierig und teuer wäre, und die BBQ nicht riesige finanzielle Mittel zur Verfügung hat, fällt unsere Wahl für das Projekt auf das Scrum Vorgehensmodell. Das Scrum-Modell ist ein flexibles Modell, dass sich durch eine Zahl X an Sprints auszeichnet, in welchen die verschiedenen Anforderungen des Product Backlogs umgesetzt werden. Genauer gesagt unterteilt sich ein solcher Sprint in verschiedene Phasen:

1. **Sprint Planning:** Der Sprint wird vorbereitet, aus dieser Phase geht der Sprint-Backlog hervor, der die Anforderungen aus dem Product Backlog enthält, welche innerhalb dieses Sprints umgesetzt werden müssen.
2. **Daily Scrum:** Der Sprint hat gestartet, es wird jeden Tag geprüft, wie weit das Projektteam gekommen ist. Bei mehr Zeitbedarf, lässt sich das Sprint-Backlog erneut anpassen. Aus dieser Phase resultiert dann das Increment, dass im Prinzip die neue Version des Produkts mit den neuen Verbesserungen/Anpassungen ist.
3. **Scrum Review:** Die Sprintergebnisse werden überprüft und mit den Stakeholdern (BBQ GmbH) abgestimmt.
4. **Sprint Retrospective:** Es wird nochmal geprüft, ob die Arbeitsweise so passt und was in Zukunft verbessert werden kann.

Im Fall der BBQ GmbH ist das Scrum Modell zu empfehlen und eignet sich hervorragend, da jederzeit Anforderungen nachgebessert werden können. Zudem lässt es uns als Dienstleister die Flexibilität, Anforderungen auch ein wenig später umzusetzen, sollte mehr Zeit benötigt werden. Des Weiteren sind wir ein kleines Dienstleistungsunternehmen, dass aus vier Personen besteht. Das Scrum Modell eignet sich hervorragend für eine kleinere Personengruppe, da das ausführende Scrum Team nur aus Product Owner, Scrum Master und Developern besteht.

2 Projektorganisation

Ein klarer Projektplan mit definierten Verantwortlichkeiten ist essentiell, damit das Scrum-Modell Erfolg haben kann. Deshalb ist zunächst einmal wichtig, dass die verschiedenen Rollen innerhalb des Projekts verteilt werden.

Rolle	Name der Person	Aufgabe
Product Owner	Manuel Lutz	Maximierung des Mehrwerts, dass durch das Produkt generiert wird
Scrum Master	Luca Schmoll	Ergebnisverantwortlich für das Scrum-Modell.

Developer	Fabian Steiß, Magnus Lörcher	In jedem Sprint nutzbare Increments schaffen
-----------	------------------------------	--

2.1 Liefergegenstände und Meilensteine

Damit der Geschäftsführung der BBQ GmbH ein erster Ansatz geboten werden kann, werden in diesem Abschnitt die Liefergegenstände und Meilensteine dokumentiert und festgehalten. Diese werden basierend auf dem Scrum Modell entwickelt.

Liefergegenstände:

Nummer	Name	Beschreibung	Verantwortlicher	Zugehöriger Meilenstein
1	Product Backlog	Ein vollständiger Product Backlog für die Geschäftsführung (GF), der durch diese abgesegnet werden kann.	Product Owner	M1
2	Projektplan	Ersteinschätzung des Projektablaufs, wie viele Phasen, Verbindlichkeiten, Kosten, etc. Soll dienen, um die GF ins Bild zu setzen.	Scrum-Master / Product Owner	M2
3	Sprint Increment 1	Eine erste Version der lauffähigen Software. Soll der GF dienen, eventuell am Product Backlog nochmals nachzubessern	Developer / Product Owner	M3
4	Sprint 1-6 Release Notes	Dokumentation der Funktionen, die innerhalb der Sprints als Increments umgesetzt wurden. Dient späteren Nutzern zur Nachverfolgung und der GF zur Übersicht, welche Funktionen umgesetzt wurden/verschoben wurden.	Scrum Master	M4

5	Sprint 6 Key User Auswertung	Auswertung der Beurteilung der Testpersonen/Fachbereiche, damit GF die Übersicht behält.	Product Owner / Scrum Master	M5
6	Go-Live	Paket mit Anleitungen und allen nötigen Informationen, um mit dem System live zu gehen.	Developer	M6

Meilensteine:

Nummer	Meilenstein	Beschreibung	Datum	Verantwortlich
M1	Product Backlog Refinement	Product Backlog vorläufig erstellt und bereit für die Review bei der GF	19.12.2024	Product Owner
M2	Initiale Sprintplanung	Erste Einschätzung des Projektumfangs getroffen, Sprint-Längen geplant, Ressourcen und Kostenplan erstellt	20.12.2024	Scrum Master, Product Owner
M3	Sprint Review – Increment 1	Erstes funktionsfähiges Product-Increment präsentiert und von GF bewerten lassen	13.01.2025	Entwickler, Product Owner
M4	Sprint-Review (Release Notes)	Zusammenfassung der entwickelten Features der ersten Sprint-Phase inklusive vollständiger Dokumentation	24.01.2025	Scrum Master
M5	Key-User-Feedback Review	Bewertung der Key-User Erfahrungen aus Sprint 6 für Anpassungen am Product-Backlog	25.04.2025	Product Owner, Scrum Master
M6	Go-Live / Product Increment	Lieferung des finalen Produkts mit vollständiger Dokumentation für den operativen Einsatz	offen	Entwickler

3 Stakeholder-Analyse

Damit der Umsetzung der Software nichts entgegen steht, ist es wichtig, alle Stakeholder zu analysieren, um später die Software an deren Bedürfnisse anzupassen. Die folgenden Stakeholder sind durch das Projekt in irgendeiner Weise betroffen:

- Geschäftsführung
 - Negativer Stakeholder: Die Geschäftsführung muss die Software kritisch betrachten. Sie müssen nach Fehlen suchen und neue Wünsche äußern. Sie können also als kritische Reflexion für das Projekt mit einbezogen werden.
- HR-Abteilung
 - Positiver Stakeholder: Die HR-Abteilung hat ein hohes Eigeninteresse an diesem Projekt. Sie ersparen sich durch dieses Projekt einiges an Arbeitszeit. Sie können zur Funktionsverbesserung genutzt werden. Außerdem können sie Ideen für das Projekt einbringen und die meisten Funktionen testen.
- IT-Infrastruktur
 - Negativer Stakeholder: Die IT-Infrastruktur ist dafür zuständig, dass am Ende alle Server laufen und aufgesetzt sind. Sie sind ebenfalls dafür zuständig, dass die Chips zum händischen Stempeln innerhalb der BBQ GmbH funktionieren. Sie werden ebenfalls kritisch auf das Projekt schauen, insbesondere auf die Möglichkeiten, die Software in IT-Systeme zu integrieren. Sie können, wie auch die Geschäftsführung, helfen, die Software zu verbessern und anzupassen, insbesondere im Bezug auf die IT-Systeme.
- Mitarbeiter anderer Abteilungen
 - Neutraler Stakeholder: Alle anderen Mitarbeiter sind nicht direkt durch das Projekt betroffen. Für diese wird es zwar am Ende eine Zeitersparnis sein, dass Zeitbuchen nicht mehr in Papierform zu machen, haben aber sonst keinen Vor oder Nachteil. Sie können für das Projekt als Tester mit einbezogen werden.

- IT-Softwareentwickler
 - Positive Stakeholder: Die Softwareentwickler haben ebenfalls ein hohes Interesse an diesem Projekt. Durch die API wird es ihnen möglich sein, dieses System in diverse andere Systeme zu integrieren, z.B. das Reisemanagement-System. Sie können für das Projekt genutzt werden, um jegliche Möglichkeiten der API-Anbindung zu besprechen.
- IT-Security
 - Negativer Stakeholder: Für IT-Security steht die Sicherheit des Projekts im Fokus. Für diese wird es wichtig sein, dass das Projekt in Bezug auf die Sicherheit absolut wasserdicht ist. Hierzu werden sie etliche Tests durchführen und die Sicherheit kritisch hinterfragen. Sie können jedoch positiv für das Projekt eingesetzt werden, da durch die Anmerkungen zur Sicherheit die Sicherheit verbessert werden kann.

4 Anforderungsanalyse

Damit die Umsetzung des Projekts beginnen kann, muss durch das Scrum Team und die zuvor benannten Stakeholder, ein Product Backlog erstellt werden. Dieses Product Backlog ist keineswegs in Stein gemeißelt, es kann und wird während des Prozesses immer wieder verändert und angepasst. Im ersten Schritt jedoch, muss eine Ist-Aufnahme und Ist-Analyse des aktuellen Prozesses durchgeführt werden. Bei der BBQ GmbH handelt es sich um ein kleines Unternehmen, wie oben bereits benannt. Der aktuelle Prozess beläuft sich bei diesen lediglich auf Papier. Jeder Mitarbeiter hat aktuell einen Zettel, auf welchem er seine gearbeiteten Zeiten einträgt. Diese werden am Ende des Monats durch die HR-Abteilung ausgewertet, damit ermittelt werden kann, ob die Person genug gearbeitet hat, oder nicht. Des Weiteren prüft die HR-Abteilung auf Überstunden oder Arbeitszeit außerhalb der Kernarbeitszeit. Es gibt also nicht viel zu analysieren und aufzunehmen. Es gilt also, ein komplett neues, alles umfassendes, System zu entwickeln, dass alle Anforderungen, die durch die verschiedenen Fachbereiche festgestellt werden, umsetzt. Damit dies verwirklicht werden kann, haben wir als Dienstleister uns mit den einzelnen Fachbereichen zusammengesetzt und deren Anforderungen besprochen. Es

sind durch diese Gespräche dann folgende Userstories entstanden, die in verschiedene Epics aufgeteilt wurden:

Ich bin Mitarbeiter der BBQ GmbH, für mich ist eine übersichtliche Software relevant, die mich über meine Fehler rechtzeitig informiert.

1. Hierzu soll die Software den Nutzer über sein Arbeitszeitmodell informieren
 2. Hierzu soll die Software dem Nutzer ermöglichen, ein doppeltes Stempeln zu vermeiden
 3. Hierzu soll die Software responsiv gestaltet werden
 4. Hierzu soll die Software verschiedene Systeme unterstützen
- ⇒ Ich bin Mitarbeiter der BBQ GmbH, für mich ist es wichtig, dass ich auf der Nutzungsoberfläche im Web alles schnell finden kann. **(8)**
1. Es wird dem Nutzer eine Navigationsleiste zur Verfügung gestellt, in welcher er die wichtigsten Funktionen findet. Zu diesen gehören die Einstellungen, das Ausloggen und die Benachrichtigungen des Systems
 2. Es gibt ein klar strukturiertes Layout, dass es jedem Nutzer ermöglicht, schnell den Überblick über die Zeiterfassung zu bekommen.
 3. Es werden Icons verwendet, damit Nutzer schneller die Funktion begreifen können.
- ⇒ Ich bin Mitarbeiter der BBQ GmbH, ich möchte gerne die Möglichkeit besitzen, meine Arbeitszeiten der vorherigen Tage zu sehen **(3)**
1. Es gibt die Möglichkeit, per Pfeiltaste innerhalb einer Tabelle, die alle Arbeitszeiten anzeigt, durch die Tage zu navigieren
 2. Es gibt die Möglichkeit, per Eingabe, ein Datum auszuwählen, an welchem Tag man seine Arbeitszeit sehen möchte.
- ⇒ Ich bin Mitarbeiter der BBQ GmbH, ich hätte gerne die Möglichkeit, die Sprache zu ändern, ohne dass ich die Software neu starten muss und lange danach suchen muss. **(2)**
1. Der Nutzer hat in den Einstellungen einen Schieberegler, der das Wechseln der Sprache ermöglicht.
 2. Der Nutzer wird, sollte er die Sprache wechseln, auf eine andere Seite weitergeleitet, ohne dass er das System neu starten muss.
- ⇒ Ich bin Mitarbeiter der BBQ GmbH, ich würde gerne die Software sowohl auf meinem Geschäftslaptop nutzen können (Windows), als auch auf meinem Rechner in der Firma (MacOS). **(1)**

1. Damit das System überall lauffähig ist, wird das System als WebApp geschrieben, es kann also nicht nur auf Windows und MacOS, sondern jeglichem System geöffnet werden.
 2. Damit es überall einigermaßen gut aussieht, wird Bootstrap als Framework verwendet, damit das Design responsiv ist.
- ⇒ Ich bin Mitarbeiter der BBQ GmbH, ich würde gerne sehen können, wie lange ich bereits gearbeitet habe und wie lange ich noch arbeiten muss. **(3)**
1. Der Nutzer bekommt auf der rechten Seite eine Übersicht zur Verfügung gestellt, in welche ihm angezeigt wird, wie lange er noch zu arbeiten hat.
 2. In dieser Übersicht wird außerdem angezeigt, wie viele Stunden er diesen Tag noch zu arbeiten hat, abgezogen der Pausen,
- ⇒ Ich bin Mitarbeiter der BBQ GmbH, ich würde gerne sehen, ob und wie viele Stunden ich außerhalb der Kernarbeitszeit von 6-22 Uhr gearbeitet habe. **(3)**
1. Der Nutzer bekommt rechts auf der Software angezeigt, wie viele Stunden er außerhalb der Kernarbeitszeit gearbeitet hat.
 2. Sollte der Nutzer keine Stunde außerhalb der Arbeitszeit gearbeitet haben, so wird dort einfach ein „-“ angezeigt.
- ⇒ Ich bin Mitarbeiter der BBQ GmbH, ich würde mir wünschen, dass meine Pausen automatisch abgezogen werden. **(5)**
1. Der Mitarbeiter bekommt am Ende des Tages alle Pausenzeiten automatisch abgezogen, die für die Anzahl an Stunden, die er gearbeitet hat, nötig gewesen wären.
 2. Der Mitarbeiter bekommt eine Nachricht, sofern er durch die abgezogenen Pausen zu wenig gearbeitet hat.
 3. Der Mitarbeiter bekommt eine Nachricht, sofern er durch die abgezogenen Pausen dennoch zu viel gearbeitet haben sollte.

Ich bin ein Teil der Geschäftsführung für mich sind die rechtlich bindenden Kriterien besonders wichtig. Ebenfalls die Umsetzung des Projekts.

1. Hierzu soll das Arbeitszeitschutzgesetz umgesetzt werden
 2. Hierzu soll das Jugendarbeitsschutzgesetz umgesetzt werden
- ⇒ Ich bin ein Teil der Geschäftsführung, ich möchte, dass die Nutzer über Vergehen gegen das Arbeitszeitschutzgesetz informiert werden. **(8)**

1. Die Nutzer bekommen eine Benachrichtigung, wenn sie mehr arbeiten, als ihr Arbeitszeitmodell vorsieht
 2. Die Nutzer bekommen eine Benachrichtigung, wenn sie weniger gearbeitet haben, als ihr Arbeitszeitmodell das vorsieht.
 3. Jeder Nutzer bekommt eine halbe Stunde Pause abgezogen, wenn er mehr als 6 Stunden und weniger als 9 Stunden gearbeitet hat
 4. Der Nutzer bekommt eine dreiviertel Stunde Pause abgezogen, wenn er mehr als 9 Stunden gearbeitet hat.
 5. Das System blockiert das Einstampeln, wenn der Nutzer noch keine 11 Stunden Pause zwischen seiner letzten Schicht hatte.
- ⇒ Ich bin ein Teil der Geschäftsführung, ich möchte, dass die Nutzer über Vergehen gegen das Jugendarbeitsschutzgesetz informiert werden. **(13)**
1. Das System benachrichtigt den Nutzer, wenn er unter 15 ist und mehr als 2 Stunden gearbeitet hat
 2. Das System benachrichtigt den Nutzer, wenn er zwischen 15 und 18 ist und mehr als 8 Stunden gearbeitet hat.
 3. Das System zieht dem Nutzer, wenn er zwischen 15 und 18 ist, automatisch 30min Pause ab, wenn er 4,6-6 Stunden gearbeitet hat.
 4. Das System zieht dem Nutzer, wenn er zwischen 15 und 18 ist, automatisch 30min Pause ab, wenn er 6-8 Stunden gearbeitet hat.
 5. Das System erinnert den Nutzer eine Stunde nach Beginn seiner Arbeitszeit und eine Stunde vor Ende seiner Arbeitszeit daran, dass er Pause machen kann.
 6. Das System prüft, dass sich ein Nutzer zwischen 15 und 18 nicht vor 6 und nach 20 Uhr einloggen kann.
 7. Das System benachrichtigt den Nutzer, wenn er zwischen 15 und 18 ist und mehr als 5 Tage in einer Woche gearbeitet hat.
- ⇒ Ich bin ein Teil der Geschäftsführung, ich möchte das die Nutzer nur in der Kernarbeitszeit arbeiten, von 6-22 Uhr. **(5)**
1. Das System informiert den Nutzer, wenn er sich vor 6 und nach 22 Uhr einloggen möchte, dass das keine Kernarbeitszeit ist
 2. Sollte der Nutzer dennoch arbeiten wollen, wird das separat zur normalen Arbeitszeit gespeichert und angezeigt
- ⇒ Ich bin ein Teil der Geschäftsführung, mein Wunsch ist es, dass meine Arbeiter sich nicht am Wochenende Einstampeln können. **(1)**

1. Das System blockiert, dass Mitarbeiter sich am Wochenende Einstempeln können.
⇒ Ich bin Teil der Geschäftsführung, für mich ist wichtig, dass alle Termine beim Projekt eingehalten werden und alles pünktlich geliefert wird. **(3)**
1. Für das Projekt wurden Meilensteine und Liefergegenstände festgesetzt.
2. Diese werden pünktlich geliefert und umgesetzt.
3. Sollte es zu Verzögerungen kommen, wird die Geschäftsführung durch den Scrum-Master umgehend informiert.
⇒ Ich bin ein Teil der Geschäftsführung, ich wünsche, dass die Nachrichten des Systems nicht deaktiviert werden können. **(1)**
1. Das System beinhaltet keine Möglichkeiten, die Nachrichten des Systems abzustellen.
⇒ Ich bin ein Teil der Geschäftsführung, ich wünsche mir, dass ich regelmäßig in die Projektumsetzung mit einbezogen werde. **(3)**
1. Die Geschäftsführung wird regelmäßig nach den Sprintphasen konsultiert und informiert.

Ich bin ein Teil der HR-Abteilung. Für mich ist es sehr wichtig, dass die Mitarbeiter künftig ihre Zeiten digital erfassen können. Das spart mir einiges an Arbeit.

1. Es muss sichergestellt sein, dass jeder Mitarbeiter individuell seine Zeiten stempeln kann.
2. Es muss sichergestellt werden, dass jeder eine Übersicht seiner gearbeiteten Zeiten hat.
3. Es muss sichergestellt werden, dass ich als HR-Mitglied eine Übersicht der gearbeiteten Zeiten eines jeden habe.
4. Es muss sichergestellt werden, dass ich als HR-Mitglied, Nutzer verwalten kann.
⇒ Als HR-Mitglied wünsche ich mir eine übersichtliche Verwaltungsoberfläche. **(8)**
1. Es muss innerhalb der Verwaltungsoberfläche klar erkennbar sein, welchen Nutzer ich gerade betrachte.
2. Es muss ein einheitliches Design vorliegen, Icons an gegebener Stelle, Übersicht immer gleich, etc.
⇒ Als HR-Mitglied wünsche ich mir die Möglichkeit, Arbeitszeiten der Mitarbeiter zu exportieren. **(5)**
1. Es muss eine Liste geben, die exportiert werden kann, um am Monatsende die Arbeitszeiten in das System für Überweisungen zu übertragen.

2. Es muss sichergestellt werden, dass nur die relevanten Informationen in dieser Liste generiert werden.
- ⇒ Als HR-Mitglied wünsche ich mir die Möglichkeit zur Nutzerverwaltung. **(8)**
1. Es muss sichergestellt werden, dass neue Nutzer angelegt werden können.
 2. Es muss sichergestellt werden, dass alte Nutzer gelöscht werden können.
 3. Es muss sichergestellt werden, dass das Passwort der Nutzer gelöscht werden kann.
 4. Es muss sichergestellt werden, dass Arbeitszeiten der Nutzer bearbeitet werden können

Ich bin Teil der Software-Entwicklung. Für mich bietet die Zeiterfassung viele Möglichkeiten, unsere bereits bestehende Software zu verbessern.

1. Es muss sichergestellt werden, dass eine API mit verschiedenen Funktionen zur Verfügung steht.
 2. Es muss sichergestellt werden, dass der Code zur Wartung/Veränderung gut lesbar ist.
 3. Es muss ein Tool zur Versionsverwaltung verwendet werden.
- ⇒ Als Software-Entwickler wünsche ich mir, dass ich mit Hilfe der API das Zeiterfassungssystem steuern kann. **(13)**
1. Es muss eine Möglichkeit geben, per API einen Mitarbeiter einzustempeln.
 2. Es muss eine Möglichkeit geben, per API einen Mitarbeiter auszustempeln.
 3. Es muss eine Möglichkeit geben, per API die Nachrichten abzurufen (Kommen, Gehen)
 4. Es muss eine Möglichkeit geben, per API die NutzerID abzurufen
 5. Es muss eine Möglichkeit geben, Nutzer über die API anzulegen und zu löschen.
- ⇒ Als Software-Entwickler wünsche ich mir, dass der Source-Code klar verständlich ist, um in Zukunft das System weiter entwickeln zu können. **(8)**
1. Es muss eine vollständige technische Dokumentation der Anwendung geben, die alle Funktionen umfasst.
 2. Es benötigt einen klar strukturierten Source-Code für das bessere Verständnis
 3. Es sollten Kommentare an den wichtigen Stellen im Source-Code vorhanden sein.
- ⇒ Als Software-Entwickler ist es wichtig, dass ich bei der Entwicklung im Zweifel auf vorherige Versionen zugreifen kann. **(3)**
1. Es muss ein System geben, dass jede Änderung zur Nachverfolgung bereitstellt.

2. Hierzu wurde GIT ausgewählt, dass die Versionierung ermöglicht.

Ich gehöre zur IT-Infrastruktur. Für mich ist primär wichtig, dass das System sich gut in bestehende IT-Systeme integrieren lässt.

1. Hierzu muss das System einfach einzurichten sein
 2. Das System sollte performant auf den Servern laufen
- ⇒ Als Mitglied der Infrastruktur ist es mir wichtig, dass ich das System einfach zum laufen bringe, ohne großartig viele, neue Server aufzusetzen. **(5)**
1. Hierzu soll ein Installationsskript mitgeliefert werden, dass die gesamte Software installiert
- ⇒ Als Mitglied der Infrastruktur ist mir wichtig, dass die Software performant läuft, damit keine anderen Systeme davon betroffen sind. **(5)**
1. Hierzu sollen unnötige Code-Ausführungen vermieden werden
 2. Es soll eine effiziente Datenbanklösung gewählt werden, in diesem Fall MariaDB

Als Sicherheitsbeauftragter habe ich die Verantwortung, sicherzustellen, dass alle personenbezogenen Daten, die durch das System verarbeitet werden, sicher verwaltet und übertragen werden.

1. Hierzu soll der Zugang zu den Systemen lokal in der Firma bleiben.
 2. Jeder Nutzer individuell erstellt werden.
 3. Für Sicherheit des Webserverns gesorgt werden.
- ⇒ Ich bin der Sicherheitsbeauftragte, für mich ist es besonders wichtig, dass das System innerhalb der Firma betrieben wird und kein Zugriff von außerhalb stattfinden kann. **(3)**
1. Hierzu soll das System lokal auf unseren Firmenrechnern betrieben werden.
 2. Es soll außerdem nur möglich sein, auf das System von außerhalb zuzugreifen (Mitarbeiter im Home-Office), wenn dieser per VPN im Firmennetzwerk verbunden ist.
- ⇒ Ich bin Sicherheitsbeauftragter, für mich ist sehr wichtig, dass jeder Nutzer individuell unterscheidbar und gesichert ist. **(5)**
1. Hierzu soll jeder Nutzer ein eigenes Passwort vergeben können, nachdem er sich zum ersten Mal angemeldet hat.
 2. Jeder Nutzer soll die Möglichkeit besitzen, dieses Passwort jederzeit in den Einstellungen zu ändern.

3. Dieses Passwort sollte eine Mindestlänge von 12 Zeichen besitzen, damit die Sicherheit gewährleistet wird.
 4. Das Passwort muss mindestens ein Sonderzeichen enthalten.
- ⇒ Ich bin Sicherheitsbeauftragter, für mich ist es wichtig, dass der Webserver über eine Verschlüsselung verfügt, damit Daten nicht im Klartext übertragen werden. **(3)**
1. Für die Weboberfläche des Systems wird eine HTTPS eingerichtet.

Diese sind in der Reihenfolge sortiert, in welcher sie ausgeführt werden sollen. Besser gesagt, wie die Priorisierung liegt. Diese Stories werden für das Projekt als Product-Backlog dienen.

5 Zeitplan und Budgetberechnung

Damit der Aufwand eingeschätzt werden kann, hat sich das Scrum Team gemeinschaftlich alle User-Stories angeschaut und ein Planning Poker durchgeführt. Hierbei hatte jedes Mitglied Karten, um den Aufwand zu bewerten. Diese Karten basieren auf der Fibonacci Folge. Je höher der Wert, desto aufwendiger ist die Umsetzung. Diese Werte können bei der Planung berücksichtigt werden. Konkret wurde die genaue Sprint-Planung und Aufgabenplanung in Trello durchgeführt, da dieses Tool das unterstützt. Der Link zu Trello findet sich im Git-Repo. Der konkrete Zeitplan für das Projekt findet sich jedoch hier:

Aufgabe	Beginn	Fertigstellung
Einteilung des Scrum-Teams / Projektbeginn	19.12.2024	20.12.2024
Verfassen der Meilensteine, der Liefergegenstände und des Product Backlogs.	20.12.2024	13.01.2025
Zeitplanung und Budgetberechnung für das Projekt	13.01.2025	24.01.2025
Sprint 1	24.01.2025	07.02.2025
Auswerten der Sprintphase – Anpassen des Projekts / Product Backlogs	07.02.2025	14.02.2025
Sprint 2	14.02.2025	28.02.2025
Auswerten der Sprintphase – Anpassen des Projekts / Product Backlogs	28.02.2025	07.03.2025

Sprint 3	07.03.2025	21.03.2025
Auswerten der Sprintphase – Anpassen des Projekts / Product Backlogs	21.03.2025	28.03.2025
Sprint 4	28.03.2025	04.04.2025
Sprint 5	28.03.2025	18.04.2025
Auswerten der Sprintphasen – Anpassen des Projekts / Product Backlogs	18.04.2025	25.04.2025
Sprint 6	25.04.2025	09.05.2025
Closed Beta für verschiedene Einzelpersonen	09.05.2025	30.05.2025
Änderungen der Fehler, die bei den Tests rauskamen	30.05.2025	10.06.2025
Erstellen des Go-Live Pakets und Unterstützung bei der Installation	10.06.2025	12.06.2025
Go-Live der finalen Software	13.06.2025	13.06.2025

Da die BBQ GmbH ein kleines Unternehmen ist, stehen ihnen für das Projekt auch nur begrenzte Mittel zur Verfügung. Sie geben für die Software ein Budget von 120000€ frei. Dieses Budget wird von uns wie folgt genutzt:

Kostenpunkt	Höhe der Ausgabe
Projektplanung	6000€
Gehalt Scrum-Master	30000€
Gehalt pro Developer	24000€
Kosten Hardware	3000€
Installation der Software vor Ort	1000€
Aufsetzen der Datenbank	500€
Wartung für 1 Jahr	12000€
Entwicklungssupport für 1 Jahr	18000€
Schulungsmaterial für Mitarbeiter	500€
Puffer / unerwartete Kosten	1000€

Es bleiben also 1000€ Puffer, die frei verfügbar, an Stellen geleitet werden können, die diese Mittel benötigen.

6 Wireframes & Prototyp

Im Rahmen der technischen Planung hat sich unser Team entschieden, die Zeiterfassungssoftware als Webanwendung umzusetzen.

Diese Lösung ermöglicht eine plattformunabhängige Nutzung – sie funktioniert problemlos auf Windows-, MacOS - und Linux-Systemen sowie auf mobilen Endgeräten.

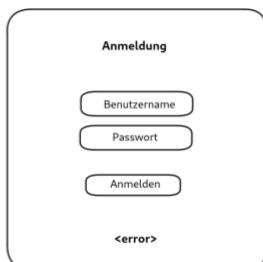
Webbasierte Anwendungen haben sich in vielen Unternehmen bewährt und bieten den Vorteil, dass sie zentral gewartet und aktualisiert werden können.

Im Vergleich zu rein softwareseitigen Installationen ist die Verwaltung deutlich einfacher und flexibler – insbesondere im Hinblick auf zukünftige Erweiterungen oder Änderungen.

Die Anwendung soll auf einem zentralen Webserver innerhalb des Firmennetzwerks der BBQ GmbH betrieben werden um eine sichere und effiziente Verwaltung zu ermöglichen.

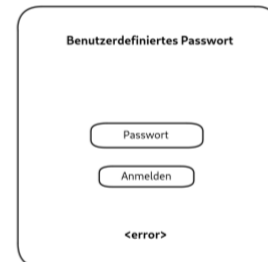
6.1 Wireframes

Login Seite



A wireframe for a login page. It features a central rounded rectangle containing the text "Anmeldung" at the top. Below it are two input fields labeled "Benutzername" and "Passwort". At the bottom of the rectangle is a button labeled "Anmelden" and a placeholder for an error message, "<error>".

Benutzerdefiniertes Passwort



A wireframe for a page titled "Benutzerdefiniertes Passwort". It contains a central rounded rectangle with the title at the top. Below the title are two input fields, the first labeled "Passwort". At the bottom of the rectangle is a button labeled "Anmelden" and a placeholder for an error message, "<error>".

User Dashboard

Benachrichtigungsfenster

The image displays two wireframe diagrams of a web application interface, likely for a time tracking or attendance system. The left diagram represents a desktop view, and the right diagram represents a mobile view.

Desktop View (Left):

- Header:** "Guten Tag <benutzername>" (Hello <username>).
- Navigation:** "Benachrichtigungen" (Notifications), "Logout", and "Einstellungen" (Settings).
- Table:** A large table with columns "Datum" (Date), "Kommen/Gehen" (Arrive/Leave), and "Uhrzeit" (Time). The table has 10 rows.
- Form Elements:** "Früher" (Earlier), "dd.mm.yyyy" (Date), and "Später" (Later) buttons.
- Annotations:** "Anwesend / Abwesend" (Present / Absent), "Soll - Arbeitszeit:" (Should - Working time:), "Stundenwoche:" (Hours per week:), and "Gleitzeitkonto:" (Flexi-time account:).
- Bottom Bar:** "→ Kommen" (Arrive) and "← Gehen" (Leave) buttons.

Mobile View (Right):



- Header:** "Guten Tag <benutzername>" (Hello <username>).
- Navigation:** "Benachrichtigungen" (Notifications), "Logout", and "Einstellungen" (Settings).
- Table:** A smaller table with columns "Datum" (Date), "Kommen/Gehen" (Arrive/Leave), and "Uhrzeit" (Time). The table has 10 rows.
- Form Elements:** "Früher" (Earlier), "dd.mm.yyyy" (Date), and "Später" (Later) buttons.
- Annotations:** "Anwesend / Abwesend" (Present / Absent), "Soll - Arbeitszeit:" (Should - Working time:), "Stundenwoche:" (Hours per week:), and "Gleitzeitkonto:" (Flexi-time account:).
- Bottom Bar:** "→ Kommen" (Arrive) and "← Gehen" (Leave) buttons.
- Mobile-Specific Elements:** A "Schließen" (Close) button, a "Stundenwoche:" (Hours per week:) label, and a "Gleitzeitkonto:" (Flexi-time account:) label.

Hamburger Menü

[illegible]

Control Panel

Guten Tag <benutzername>

Benutzername	Status	Soll - Arbeitszeit	Minderjährig	Sprache	
					Entfernen
					Entfernen
					Entfernen
					Entfernen
					Entfernen
			Übersicht Benutzer		
			Benutzer hinzufügen		
Benutzername	Passwort	Soll-Arbeitszeit Stundenwoche	Geburtstag	Sprache	Hinzufügen

7 Projektcontrolling

Für die Verwaltung aller Aufgaben, haben wir innerhalb unseres Teams auf die Software Trello zurückgegriffen. Hier haben wir alle Aufgaben, die während des Projekts anstanden angelegt und diese an uns verteilt. So war zu jeder Zeit klar, wer was zu erledigen hat. Außerdem haben wir das gesamte Product Backlog in Trello übernommen. So konnte sehr effektiv jede Sprintphase geplant werden. Der Link zu Trello findet sich im Git-Repo.

8 Qualitätssicherung

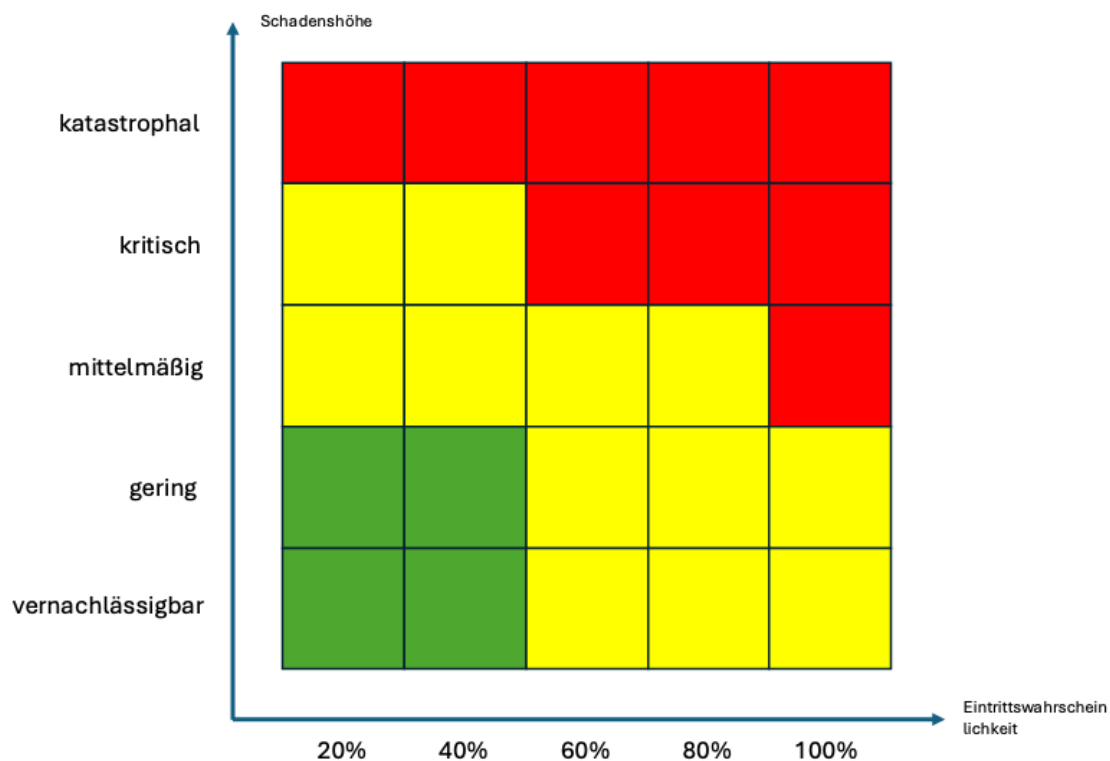
Wir als Dienstleister führen verschiedene Tests aus, um zu gewährleisten, dass am Ende einem Go-Live nichts entgegensteht. Hier zu greifen wir hauptsächlich auf Integrationstests zurück. Diese sollen, sofern möglich, nach jeder Sprintphase durchgeführt werden. Durch diese Tests sollen die neuen Module, die innerhalb dieser Sprint-Phasen entwickelt wurden, geprüft werden. So werden offensichtliche Bugs behoben und die Software verbessert. Sobald die Grundstruktur des Projekts steht, ungefähr nach 2 Sprint-Phasen, sollen zusätzlich Systemtests hinzukommen. Diese werden vom gesamten Scrum-Team gemeinschaftlich durchgeführt und sollen die gesamte Software in einem prüfen. Hier sollen Fehler entdeckt und behoben werden, die durch verknüpfte Komponenten auftreten können. Ebenfalls sollen solche betrachtet werden, die nur im Kontext der gesamten Software auftreten können. Zum Schluss soll ein großer Abnahmetest stattfinden. Dieser soll final dazu dienen, die letzten Fehler zu finden und zu beheben. Dieser Test soll in einzelnen Fachabteilungen von einzelnen Personen vor dem Go-Live durchgeführt werden. Sofern hier gravierende Fehler gefunden werden, wird ein Sprint geplant, um diese noch vor dem Go-Live zu beheben. Neben diesem großen Abnahmetest, sollen noch Performance-Tests durchgeführt werden. Hier bedarf es keiner konkreten Planung, da diese einfach nebenbei durchgeführt werden. Sicherheitstests und Nutzbarkeitstest werden nach dem Go-Live durchgeführt. Die vor dem Go-Live durchgeführten Tests werden hier dokumentiert:

Titel	Testschritte	Erwartetes Ergebnis	Status
API-Test durch Postman	Entwickler gibt Aufruf in Postman ein Postman gibt Antwort des Servers zurück	API-Calls funktionieren fehlerfrei	pass
Buttons Vor-Zurück für Wochenübersicht	Nutzer loggt sich ein Nutzer drückt jeweils den Knopf	Server druckt dem User die richtigen Daten auf dem Bildschirm	pass
User-Login	Nutzer wird im Admin-Panel angelegt Nutzer loggt sich ein Nutzer versucht sein Passwort zu ändern	Das Anlegen des Nutzers sollte einen Datenbankeintrag erstellen, dem Nutzer sollte es dann möglich sein, sich in das System.	pass

		einzuloggen. Nach erfolgreichem Login, sollte der Nutzer sein Passwort ändern können	
Funktionalitätstest	Nutzer loggt sich ein Nutzer bedient jedes Feld, dass interaktiv ist. Nutzer prüft jede Funktion	Alle Knöpfe sollten alle Interaktionen fehlerfrei durchführen.	Fail Urlaubsbutton funktioniert nicht.

9 Risikomanagement

Die Geschäftsführung der BBQ GmbH hat darum gebeten, alle Aspekte, die auftreten können und das Projekt in irgendeiner Weise gefährden könnten, aufzuführen und zu bewerten. Wir als Dienstleister greifen hier gerne auf eine Risikomatrix zurück, die uns eine einfache Bewertung ermöglicht. Diese Risikomatrix sieht wie folgt aus:



Basierend auf dieser Risikomatrix haben wir eine Abschätzung aller Aspekte vorgenommen und diese anschließend bewertet.

Risiko	Beschreibung	Eintrittswahrscheinlichkeit	Schadenshöhe	Bewertung	Zeitliche Verschiebung	Maßnahmen
Umwelteinflüsse	Feuer, Hochwasser, o.ä. zerstören	20%	katastrophal	rot	Projektabbruch	Brandschutzsysteme werden verwendet, Versicherungen werden abgeschlossen.
Krankheit Scrum-Team	Ein Projektmitglied fällt längerfristig aus	60%	mittelmäßig	gelb	Je nach Länge der Krankheit, einige Tage/Wochen	Bei Krankheit müssen die Sprints ggf. umgeplant werden.
Budgetüberschreitung	Das Budget des Projekts wird stark überschritten	80%	kritisch	rot	Projektabbruch	Die Kosten müssen regelmäßig überprüft werden, bei annähern an Budgetgrenze sollte die GF einbezogen werden
Längere Entwicklungszeit	Die Anforderungen können nicht in den geplanten Sprints umgesetzt werden	80%	gering	gelb	Einige Tage / Wochen	Sollte bemerkt werden, dass die Entwicklungszeit innerhalb der Sprints länger dauert, muss eventuell noch ein Entwickler angestellt werden / die einzelnen Sprints angepasst werden
Cyberangriff	Die Server werden gehackt, Daten gehen verloren	20%	katastrophal	rot	Projektabbruch	Server werden verschlüsselt, es werden sichere Passwörter verwendet
Unerwartete Bugs	Bugs in der Software verlängern die Entwicklungszeit	80%	gering	gelb	Einige Tage/Wochen	Sofern unerwartete Bugs auftreten, muss ein neuer Sprint geplant werden
Software nicht benötigt	Die BBQ GmbH	20%	katastrophal	rot	Projektabbruch	Regelmäßiger Austausch mit GF

	entscheidet sich, eine andere Software zu verwenden					
Lieferzeiten für Hardware	Die Hardware für die Server benötigt lange zum Liefern.	20%	vernachlässigbar	grün	Einige Tage	Bereits weit im Voraus wird die Hardware bestellt, um dieses Risiko zu minimieren.
Bankrott BBQ GmbH	Die BBQ GmbH geht bankrott.	20%	katastrophal	rot	Projektabbruch	Regelmäßiger Austausch mit GF.
Neue Anforderungen	Die Anforderungen an das Projekt ändern sich grundlegend	40%	mittelmäßig	gelb	Einige Tage / Wochen	Die Anforderungen werden nach jedem Sprint erneut besprochen, sodass nicht ein unerwartet hohes Aufkommen an neuen Anforderungen auftritt.

10 Kommunikationskonzept

Zu Beginn des Projekts haben wir als Dienstleister eine Vereinbarung mit der GF getroffen. Es ist wichtig, dass durch das Kommunikationskonzept effizient und transparent, umfassend und kontinuierlich, Informationen zum Projekt an die wichtigsten Stakeholder übermittelt werden. Deshalb haben wir zunächst vereinbart, dass zu Projektstart an alle Mitarbeiter der BBQ GmbH eine Mail versandt wird, die über das Projekt aufklärt und die ersten Informationen dazu liefert. Dies ist wichtig, da dadurch alle Fachabteilungen auf das Projekt eingestimmt werden und für eventuelle Tests bereitstehen. Nach dieser Mail wurde ein konkreter Zeitplan für die betroffenen Stakeholder festgelegt. Dieser definiert wie häufig und über was diese informiert werden müssen:

Stakeholder	Was wird kommuniziert?	Häufigkeit	Wie wird kommuniziert?	Verantwortlicher
Geschäftsführung	Ergebnisse Sprints, Wichtige Entscheidungen,	2x wöchentlich	Per Meeting und Mail	Scrum Master und Product Owner

	Dokumentationen, Zeitplanänderungen , Product-Backlog Änderungen			
HR-Abteilung	Stand der Software, potenzielle Änderungen am Product Backlog	1x alle zwei Wochen	Per Meeting	Scrum Master und Product Owner
IT-Infrastruktur	Geänderte Software- Anforderungen	1x monatlich	Per Mail	Scrum Master und Entwickler
IT-Entwickler	Änderungen an der API, wesentliche Änderungen am Source Code	1x monatlich	Per Meeting und Mail	Entwickler
IT-Security	Stand der Sicherheit im Projekt	1x monatlich	Per Mail	Scrum-Master
Mitarbeiter	Newsletter über den aktuellen Stand der Software	1x monatlich	Per Mail	Product-Owner

Basierend auf diesem Konzept wird sichergestellt, dass alle in der BBQ GmbH zu jedem Zeitpunkt über die für sie relevanten Themen informiert sind und gegebenenfalls reagieren können, falls hier Bedarf besteht.

11 Schulungskonzept

In Abstimmung mit der Geschäftsführung wurde zum Zweck der Schulung eine PowerPoint-Präsentation entwickelt. Diese Präsentation soll verpflichtend für jeden Mitarbeiter stattfinden, der neu ins Unternehmen kommt. Sie wird dann vom Product Owner durchgeführt. Zum Release der Software wird dieser Vortrag ebenfalls einmal gehalten. Hier wird die gemeinschaftlich durch den Scrum-Master und den Product-Owner gehalten. Innerhalb dieser Präsentationen sollen die wichtigsten Funktionen der Software aufgeführt und erklärt werden. Sie soll im Nachgang als eine Art Nachschlagwerk dienen, sodass die Nutzer immer Rat zur Seite haben. Des Weiteren soll während des Vortrags eine Live-Demo stattfinden. Hier zu soll ein Test-Nutzer angelegt werden, der durch jede Person angemeldet werden kann. So können die Nutzer die alle Funktionen selbst erkunden und ausprobieren. Des Weiteren wurde ein Konzept, ähnlich zu dem des Kommunikationskonzepts entwickelt. Dies kommt daher, dass es einige Stakeholder gibt, die eine andere Sicht der Software benötigen:

Stakeholder	Art der Schulung	Inhalt der Schulung	Häufigkeit	Verantwortlich
IT-Software Entwickler	Präsentation, Demo	Die Software-Entwickler bedürfen einer intensiven Schulung, die über die gesamte Software im inneren aufklärt. Das bedeutet, ihnen wird intensiv der Source Code vorgestellt. Ebenfalls wird ihnen intensiv vorgestellt, wie die API arbeitet und zu benutzen ist, sodass diese die Software nutzen/erweitern können.	1x bei Release, 1x jährlich für neue Software-Entwickler	Bei Release: Entwickler Nach Release: Product Owner
IT-Infrastruktur	Schulung vor Ort	Die IT-Infrastruktur muss speziell geschult werden, damit sie wissen, welche Hardware sie für die Software benötigen und wie diese zu installieren ist. Außerdem müssen sie geschult werden, welche	1x vor Release 1x jährlich nach Release	Vor Release: Entwickler Nach Release: Product Owner

		Zusatzsoftware noch installiert werden muss.		
HR-Abteilung	Präsentation	Da die HR Abteilung, zusammen mit der GF, die Verwaltung übernimmt, bedarf es für diese, zur Schulung für alle Mitarbeiter, eine zusätzliche Schulung für das Admin-Panel. Dieses soll ihnen erklärt werden, sodass die HR-Abteilung künftig Nutzer anlegen, ändern, löschen, etc. kann.	2x vor Release 1x im Quartal	Product Owner / Scrum Master
Geschäftsführung	Präsentation vor Ort	Die Geschäftsführung ist das zentrale Verwaltungselement. Deshalb muss sie über alles informiert werden. Es wird deshalb ein Vortrag entwickelt, der alle oben erwähnten Themen abdeckt, jedoch verkürzt. Das ermöglicht, dass die Geschäftsführung handlungs- und entscheidungsfähig bleibt.	1x vor Release Nach Bedarf	Product Owner, Scrum Master, Entwickler

Dieses Konzept stellt sicher, dass alle Mitarbeiter bestens vorbereitet sind, angepasst an ihre Rollen.

12 Beistellungen BBQ GmbH

Damit die Software durch uns bereitgestellt werden kann, benötigen wir einen fertig eingerichteten Linux-Server. Auf diesem werden dann durch uns, die Datenbanken und Systeme installiert, die benötigt werden, um die Software lauffähig zu machen. Außerdem

muss die BBQ GmbH die Arbeitszeit von einzelnen Nutzern bereit stellen, damit die Tests anständig durchgeführt werden können.

13 Erweiterungspotenzial

Die Software, die durch uns entwickelt wurde, bietet in vielerlei Hinsicht Möglichkeiten, diese zu erweitern. So ist es ohne Probleme möglich, ein Urlaubssystem mit zu integrieren. Innerhalb dieses Systems kann der Nutzer dann flexibel seinen Urlaub anlegen und genehmigen lassen. Außerdem ist es ohne größere Probleme möglich, eine Referenz einzubauen, das dir zeigt, ob du im Home-Office arbeitest oder Vor Ort bist. Es lassen sich zudem noch einige Sicherheitssysteme integrieren, wie zum Beispiel eine 2FA-Authentifizierung.

14 Reflexion & Lessons Learned

Rückblickend betrachtet, hätte im Projekt einiges anders laufen können. Das Scrum-Modell war für uns eher ungeeignet. Die Einhaltung der einzelnen Sprints fiel uns schwer, da hier kein „konkreter“ Zeitpunkt feststand, wann etwas fertig sein muss. Des Weiteren war die konkrete Planung am Anfang viel zu schwammig. Was jedoch für Scrum gesprochen hat ist, dass uns immer mal wieder neue Funktionen eingefallen sind. Das lies sich relativ leicht anpassen, ohne die gesamte Planung neu machen zu müssen. Geeignet wäre vermutlich ein Modell wie Wasserfall-Scrum gewesen, da hier Elemente aus beiden Vorgehensmodellen genutzt werden. Konkret wäre es mit diesem Modell einfacher gewesen, die vorab Planung durchzuführen, da diese auf dem Wasserfall-Modell basiert. Im Wasserfall-Modell wird der Beginn des Projekts wesentlich strukturierter und umfangreicher geplant, als im Scrum-Modell, durch Kostenrechnungen, Aufwandsschätzungen, etc. Ebenfalls nicht so gut lief das Testing. Das wurde durch uns ein wenig vernachlässigt, hätte aber einen wesentlich größeren Bestandteil spielen müssen. Es wäre mit guten Tests wesentlich einfacher gewesen, die Software zu verbessern. Zudem wäre es gut gewesen, nicht ein Mitglied während des Projekts zu verlieren. Das hat die gesamte Planung über den Haufen geworfen und dazu geführt, dass vieles neu geplant werden musste. Außerdem war es eine Mehrbelastung für alle anderen Mitglieder.