

Glossar

Erstellt von: Noah Schwenk	Überprüft von: Miguel Friesen
--------------------------------------	---

Version	Effektiv ab	Beschreibung / Änderungen
1.0	24.03.2026	Erstellung
2.0	05.04.2026	Aktualisieren des Glossars

Inhalt

1. Fachliche Begriffe	1
1.1. Arbeitsplan	1
1.2. Aufgabe / Task	1
1.3. Aufgabenabhängigkeit / Task Dependency	1
1.4. Blocker	1
1.5. Clash Management	1
1.6. E-Mail-Host.....	1
1.7. Energiebedarf.....	1
1.8. Fehlplanung.....	2
1.9. Fokusphase / Mindest-Zeitblock.....	2
1.10. Globaler Administrator.....	2
1.11. Individuelles Arbeitsprofil.....	2
1.12. Kognitive Kapazität.....	2
1.13. Mandantenfähigkeit	2
1.14. Mehrfachzugehörigkeit	2
1.15. Organisierende (Organisationsadmin).....	2
1.16. Planungsalgorithmus	2
1.17. Task-Switching	3
1.18. Belastungssteuerung.....	3
1.19. Chunks (Aufgabenblöcke)	3
1.20. Echtzeit-Rescheduling.....	3
1.21. Kognitives Task-Switching	3
1.22. Regenerationsphasen	3
1.23. Soll-Fortschritt (planbasiert)	3
1.24. Train of Thought (Gedankengang)	3
1.25. Unverrückbare Termine.....	3
1.26. Abwesenheitsmanagement / Ausfallzeiten	4
1.27. Flex-Block / Flex-Pool	4
1.28. Arbeitszeit-Blöcke.....	4
1.29. Erledigt-Markierung / Manuelles Abhaken.....	4
1.30. Exklusivität / Exklusiver Block	4
1.31. Gesamtübersicht	4

1.32.	Konfliktprävention / Überschneidungs-Schutz	4
1.33.	Manuelles Verschieben	4
1.34.	Pflichtparameter	4
1.35.	Rahmenarbeitszeit	4
1.36.	Arbeitszeit-Blöcke	5
1.37.	Sperrzonen	5
1.38.	Strategy Pattern	5
1.39.	Dependency Injection	5
2.	Technische Begriffe	5
2.1.	Cloud Firestore	5
2.2.	Firebase Authentication	5
2.3.	Google Cloud Platform (GCP)	5
2.4.	Heatmap	5
2.5.	SaaS (Software as a Service)	5
2.6.	Cloud-Persistenz / Daten-Synchronität	6
2.7.	Compliance	6
2.8.	CRUD (Create, Read, Update, Delete)	6
2.9.	Drag & Drop	6
2.10.	Hardcoded (hart verdrahtet)	6
2.11.	KI-Module	6
2.12.	Abnahmekriterium / Systemtestfälle	6
2.13.	Architekturvorereitung	6
2.14.	ASAP-Status / Hard Deadline	6
2.15.	Availability (Verfügbarkeit)	7
2.16.	Browser-Kompatibilität	7
2.17.	Chunking (in Blöcke)	7
2.18.	Datensicherheit / Privatsphäre	7
2.19.	Lastspitzen	7
2.20.	Premium-Versionen	7
2.21.	Rescheduling / Auto-Rescheduling	7
2.22.	UI-Frameworks	7
2.23.	User Experience (Zielwert)	7
2.24.	CSS-Modules	7

2.25. W3C (Web-Standards)	7
2.26. Identity Management	8
2.27. SDK (Server-seitige Entwicklungs-Schnittstelle)	8
2.28. SDK (Server-seitige Entwicklungs-Schnittstelle)	8

1. Fachliche Begriffe

1.1. Arbeitsplan

Ein vom System automatisch generierter Plan, der offene Aufgaben auf konkrete verfügbare Zeitfenster verteilt. Die Generierung basiert nicht auf einer manuellen Einteilung durch den Nutzenden, sondern erfolgt algorithmisch unter Berücksichtigung von Deadlines, Prioritäten, Energiebedarf, Aufgabenabhängigkeiten, Fokusphasen und dem individuellen Arbeitsprofil.

1.2. Aufgabe / Task

Eine vom Nutzenden angelegte Arbeitseinheit im System. Eine Aufgabe enthält zwingend Titel, Beschreibung, geschätzte Bearbeitungsdauer und Deadline. Zusätzlich können Priorität, Energiebedarf, Fokusphase sowie Abhängigkeiten zu anderen Aufgaben erfasst werden, die der Planungsalgorithmus bei der Erstellung des Arbeitsplans berücksichtigt.

1.3. Aufgabenabhängigkeit / Task Dependency

Eine fachliche Beziehung zwischen zwei oder mehr Aufgaben, die festlegt, dass eine Aufgabe erst nach einer anderen begonnen werden kann. Der Planungsalgorithmus muss diese Abhängigkeiten bei der Reihenfolge der Einplanung berücksichtigen.

1.4. Blocker

Ein eingetragener Zeitraum, in dem ein Nutzender nicht für die Bearbeitung von Aufgaben verfügbar ist (z. B. Meetings, Urlaub, Krankheit). Blocker werden vom Planungsalgorithmus als unverrückbare Planungsgrenzen behandelt und niemals mit Aufgaben belegt.

1.5. Clash Management

Das Verfahren, mit dem das System auf eine nicht auflösbare Planungssituation reagiert. Kann eine Deadline aufgrund fehlender Kapazität oder Blocker nicht eingehalten werden, stoppt das System den Planungsprozess und stellt die Konflikte transparent für den Nutzenden dar (z. B. „Deadline nicht haltbar wegen Blocker“), anstatt einen fehlerhaften Plan zu erzeugen.

1.6. E-Mail-Host

Die Domain einer E-Mail-Adresse (z. B. „dhbw-stuttgart.de“), die einer Organisation im System hinterlegt ist. Eine Registrierung ist nur möglich, wenn die verwendete E-Mail-Adresse einem der zugelassenen E-Mail-Hosts der jeweiligen Organisation entspricht oder eine direkte Einladung durch einen Organisierenden vorliegt.

1.7. Energiebedarf

Ein vom Nutzenden pro Aufgabe festgelegter Wert, der die mentale Beanspruchung durch diese Aufgabe beschreibt. Der Energiebedarf definiert nicht die zeitliche Dauer, sondern gibt dem Planungsalgorithmus an, wie ressourcenintensiv eine Aufgabe ist, um kognitive Überlastung durch zu viele anspruchsvolle Aufgaben in Folge zu verhindern.

1.8. Fehlplanung

Eine Planungssituation, in der die verfügbare Kapazität eines Nutzens nicht ausreicht, um alle Aufgaben vor ihren Deadlines abzuschließen. Das System reagiert auf eine erkannte Fehlplanung mit dem Clash Management und verhindert damit die Erstellung eines unrealistischen Arbeitsplans.

1.9. Fokusphase / Mindest-Zeitblock

Die kleinste zusammenhängende Zeitspanne, in der eine Aufgabe sinnvoll bearbeitet werden soll (z. B. „mindestens 1 Stunde am Stück“). Dieser Wert wird pro Aufgabe festgelegt und verhindert, dass der Algorithmus eine Aufgabe in zu kleine, kaum nutzbare Zeitfragmente aufteilt.

1.10. Globaler Administrator

Ein systemweit privilegierter Akteur, der neue Organisationen im System registrieren und verwalten kann. Der Globale Administrator ist der Betreiber der Plattform und hat keine Einsicht in die persönlichen Aufgaben oder Planungen einzelner Nutzer.

1.11. Individuelles Arbeitsprofil

Die persönliche Konfigurationseinheit eines Nutzers, die die Rahmenbedingungen für die algorithmische Planung festlegt. Dazu gehören die verfügbare Arbeitszeit pro Tag oder Woche, die maximale Dauer zusammenhängender Arbeitsphasen, Anzahl und Dauer von Pausen sowie organisationsspezifische Arbeitszeitfenster. Jede Änderung am Profil wirkt sich auf alle zukünftig generierten Arbeitspläne aus.

1.12. Kognitive Kapazität

Die biologisch begrenzte Fähigkeit eines Menschen zu konzentrierter Wissensarbeit, die laut wissenschaftlicher Forschung auf etwa 2–4 Stunden pro Tag begrenzt ist. Das System behandelt diesen Wert als harte Planungsgrenze und verhindert algorithmisch, dass diese Kapazität bei der Generierung von Arbeitsplänen überschritten wird.

1.13. Mandantenfähigkeit

Die systemische Eigenschaft, mehrere unabhängige Organisationen gleichzeitig und datentechnisch vollständig getrennt voneinander auf derselben Plattform zu betreiben. Mandantenfähigkeit ist ein Muss-Ziel der Basis-Applikation.

1.14. Mehrfachzugehörigkeit

Die Fähigkeit eines einzelnen Benutzerkontos, gleichzeitig mehreren unabhängigen Organisationen zugeordnet zu sein.

1.15. Organisierende (Organisationsadmin)

Ein Akteur mit administrativen Rechten innerhalb einer bestimmten Organisation. Organisierende können Mitglieder in ihre Organisation einladen und ihnen organisationsinterne Rollen zuweisen, haben jedoch keine Einsicht in die persönlichen Aufgaben oder den Arbeitsplan der Nutzer.

1.16. Planungsalgorithmus

Der fachliche Kernmechanismus des Systems, der auf Basis aller erfassten

Parameter automatisch einen Arbeitsplan erstellt. Verarbeitete Parameter sind unter anderem Deadlines, Prioritäten, Energiebedarf, Aufgabenabhängigkeiten, Fokusphasen, Blocker und das individuelle Arbeitsprofil. Bei nicht lösbaren Planungskonflikten greift das Clash Management.

1.17. Task-Switching

Der psychologische und zeitliche Aufwand, der durch den Wechsel zwischen Aufgaben unterschiedlicher Art oder Kontexte entsteht. Das System minimiert Task-Switching algorithmisch, indem inhaltlich zusammengehörende Aufgaben bevorzugt in zusammenhängenden Blöcken eingeplant werden.

1.18. Belastungssteuerung

Der automatische Wechsel zwischen schweren und "stupiden" (leichten) Aufgaben zur kognitiven Entlastung des Nutzens.

1.19. Chunks (Aufgabenblöcke)

Die konkreten, vom System terminierten Zeitfragmente einer Aufgabe, die inklusive der exakten Start- und Endzeitpunkte im Arbeitsplan gespeichert werden.

1.20. Echtzeit-Rescheduling

Die Fähigkeit des Systems, bei Änderungen der Nutzerverfügbarkeit oder des Aufgabenbestands den Arbeitsplan umgehend neu zu berechnen, wofür eine sofortige Datensynchronität vorausgesetzt wird.

1.21. Kognitives Task-Switching

Ein Planungsprinzip, bei dem das System erzwingt, dass auf kognitiv anspruchsvolle Aufgaben automatisch leichtere Aufgaben folgen müssen, um Überlastung zu vermeiden.

1.22. Regenerationsphasen

Ausreichende Erholungszeiten, die nach Aufgaben mit hohem Energiebedarf zwingend eingeplant werden müssen.

1.23. Soll-Fortschritt (planbasiert)

Eine Statusanzeige beim Einsehen von Aufgaben, die ausgibt, wie viel Zeit nach dem aktuellen Plan bereits in eine Aufgabe investiert sein sollte (z. B. "Nach aktuellem Plan sollten bereits 80% der Zeit investiert sein").

1.24. Train of Thought (Gedankengang)

Der kognitive Arbeitsfluss (Flow) eines Nutzens, dessen Unterbrechung durch zu kleinteilige Aufgabenstückelung (Splitting) vom System verhindert werden soll.

1.25. Unverrückbare Termine

Aufgaben, die als feste, statische Termine (z. B. wiederkehrend jeden Freitag) definiert sind und vom Planungsalgorithmus absolut nicht verschoben werden dürfen.

1.26. Abwesenheitsmanagement / Ausfallzeiten

Gezielte Erfassung von Zeiten (z. B. Urlaub, Krankheit, private Termine), in denen nicht gearbeitet wird.

1.27. Flex-Block / Flex-Pool

Arbeitszeit-Blöcke für dynamische Aufgaben, die für alle Organisationen freigegeben sind. In diesen Flex-Blöcken ist der Algorithmus berechtigt, dynamische Aufgaben aus jeder verknüpften Organisation einzuplanen, um die Auslastung flexibel zu optimieren.

1.28. Arbeitszeit-Blöcke

Vom Nutzenden im System angelegte zeitliche Slots, die die persönliche verfügbare Arbeitszeit definieren.

1.29. Erledigt-Markierung / Manuelles Abhaken

Ein sichtbares UI-Element (Button) an der Benutzungsschnittstelle, mit dem Nutzende eine Aufgabe abschließen können. Das Abhaken löst im System ein sofortiges Rescheduling aus.

1.30. Exklusivität / Exklusiver Block

Ein Arbeitszeit-Block, der speziell für eine bestimmte Organisation reserviert ist, sodass der Algorithmus dort ausschließlich Aufgaben dieser Organisation einplanen darf.

1.31. Gesamtübersicht

Eine einheitliche Planungsansicht, in der sämtliche Aufgaben, bei Freelancer-Accounts auch organisationsübergreifend, in einer gemeinsamen, zentralen Übersicht gebündelt dargestellt werden.

1.32. Konfliktprävention / Überschneidungs-Schutz

Eine Systemlogik, die direkt bei der Speicherung sicherstellt, dass sich Arbeitszeit-Blöcke oder Blocker gegenseitig zeitlich zwingend nicht überschneiden dürfen. Das System blockiert die Speicherung sofort bei erkannten Konflikten.

1.33. Manuelles Verschieben

Eine Aktion des Nutzenden, bei der eine zuvor dynamische Aufgabe händisch an einen anderen Zeitpunkt geschoben wird, wodurch das System diese in eine statische/einmalige Aufgabe umwandelt und den restlichen Plan anpasst.

1.34. Pflichtparameter

Zwingend erforderliche Eingaben bei der Erstellung einer Aufgabe, zu denen Titel, Beschreibung, Dauer, Deadline, Priorität, Energiebedarf und Abhängigkeiten gehören.

1.35. Rahmenarbeitszeit

Die Summe aller angelegten Arbeitszeit-Blöcke, welche die absolute zeitliche Obergrenze für die algorithmische Aufgabenplanung bildet.

1.36. Arbeitszeit-Blöcke

Vom Nutzenden im System angelegte, überschneidungsfreie zeitliche Slots, die die persönliche, verfügbare Arbeitszeit für die Aufgabenplanung definieren.

1.37. Sperrzonen

Die Funktion von Blockern als harte Zeiten, in denen vom Planungsalgorithmus absolut keine Aufgaben bearbeitet werden können.

1.38. Strategy Pattern

Eine saubere, flexible Software-Schnittstelle, in die der Planungsalgorithmus eingehängt wird, sodass er nicht hart verdrahtet ist und später modular gegen komplexe Premium-Versionen oder KI-Module ausgetauscht werden kann.

1.39. Dependency Injection

Ein Software-Entwurfsmuster zur Einbindung von Abhängigkeiten von außen, das genutzt wird, um dem System beispielsweise eine simple, deterministische Strategie für den Prototyp zu injizieren und die Architektur testbar zu machen.

2. Technische Begriffe

2.1. Cloud Firestore

Eine dokumentenorientierte NoSQL-Datenbank der Google Cloud Platform. Sie dient im Projekt als primäre Datenhaltung für nutzerspezifische Strukturen wie Aufgaben, Arbeitsprofile und Arbeitspläne und bietet eine integrierte Echtzeitsynchronisation zwischen Backend und Frontend.

2.2. Firebase Authentication

Ein Dienst der Google Cloud Platform zur sicheren Benutzerauthentifizierung. Das System lagert die Verwaltung von Anmeldedaten, Passwortsicherheit und Zugriffstoken vollständig an diesen Dienst aus, um den Implementierungsaufwand zu reduzieren und Sicherheitsstandards zuverlässig einzuhalten.

2.3. Google Cloud Platform (GCP)

Die Cloud-Infrastruktur, auf der das System betrieben wird. GCP stellt im Projekt die Dienste Cloud Run (Backend-Hosting), Cloud Firestore (Datenhaltung) und Firebase Authentication (Authentifizierung) bereit.

2.4. Heatmap

Eine grafische Darstellung, die Werte durch Farbintensität visualisiert. Im Rahmen der optionalen Kapazitätsanalyse (LF70) soll eine Heatmap die zeitliche Auslastung des Nutzenden übersichtlich darstellen, um drohende Überlastungen frühzeitig erkennbar zu machen.

2.5. SaaS (Software as a Service)

Ein Bereitstellungsmodell, bei dem Software nicht lokal installiert, sondern vollständig über das Internet bereitgestellt wird. Das System folgt diesem Modell als

plattformunabhängige Web-Applikation, die ohne Installation in jedem gängigen Browser lauffähig ist.

2.6. Cloud-Persistenz / Daten-Synchronität

Die technische Anforderung, dass Änderungen an Daten innerhalb von maximal 2 Sekunden in der Cloud-Datenbank gespeichert und für sofortige nachfolgende Berechnungen bereitgestellt werden müssen.

2.7. Compliance

Die Einhaltung gesetzlicher und unternehmensinterner Richtlinien, die im Projekt explizit erfordert, dass alle Server der Infrastruktur zwingend innerhalb der Europäischen Union (EU) stehen müssen.

2.8. CRUD (Create, Read, Update, Delete)

Ein Akronym für triviale Datenverwaltungs-Applikationen, von denen sich das geplante System als hochkomplexe, reaktionsschnelle "Planungsmaschine" explizit abgrenzt.

2.9. Drag & Drop

Ein interaktives Bedienkonzept an der Benutzeroberfläche, für das unzulässige Aktionen (z.B. falsches Verschieben von Tasks) direkt durch simple Fehlermeldungen abgefangen werden müssen.

2.10. Hardcoded (hart verdrahtet)

Ein softwaretechnisches Anti-Pattern, bei dem Logik fest im Quellcode verankert ist. Im Projekt wird dies beim Planungsalgorithmus strikt vermieden (z.B. durch das Strategy Pattern), um spätere Austausche zu ermöglichen.

2.11. KI-Module

Zukünftige, auf Künstlicher Intelligenz basierende Softwarekomponenten, für die die Systemarchitektur bereits im aktuellen Prototyp über modulare Schnittstellen vorbereitet sein muss.

2.12. Abnahmekriterium / Systemtestfälle

Fest definierte Testszenarien (z. B. ein Szenario mit 50 Tasks), mit denen bei der finalen Abnahme verifiziert wird, ob die geforderten Performance-Werte eingehalten werden.

2.13. Architekturvoreereitung

Die architektonische Vorgabe, den Planungsalgorithmus nicht starr (hardcoded) zu implementieren, sondern über Schnittstellen so vorzubereiten, dass er später modular ausgetauscht werden kann.

2.14. ASAP-Status / Hard Deadline

Spezifische Priorisierungsparameter der Engine, bei denen eilige Aufgaben als Erstes eingeplant (Override) oder vor Ablauf einer knappen Deadline zwingend erzwungen werden.

2.15. Availability (Verfügbarkeit)

Der Prozess des Algorithmus, vor der Einplanung von Aufgaben die freien Slots im Kalender abzufragen.

2.16. Browser-Kompatibilität

Die nichtfunktionale Anforderung, dass das System in aktuellen Versionen von Chrome, Firefox, Edge und Safari ohne proprietäre Plugins lauffähig sein muss.

2.17. Chunking (in Blöcke)

Der technische Vorgang, bei dem die Engine eine Aufgabe in kleinere Blöcke zerteilt, wenn der im Kalender gefundene freie Slot kleiner ist als die geschätzte Aufgabendauer.

2.18. Datensicherheit / Privatsphäre

Der systemarchitektonische Schutzmechanismus, der strikt ausschließt, dass Administratoren oder Teamleiter Einsicht in die persönlichen Pläne der ihnen zugeordneten Nutzenden erhalten.

2.19. Lastspitzen

Kurzzeitige, hohe Belastungen der Cloud-Umgebung, für die das System dennoch die maximal erlaubte Antwortzeit von 20 Sekunden garantieren muss.

2.20. Premium-Versionen

Zukünftige, komplexere Ausbaustufen des Planungsalgorithmus (wie z. B. KI-Module), für deren spätere Integration die Softwarearchitektur bereits vorbereitet sein muss.

2.21. Rescheduling / Auto-Rescheduling

Die sofortige, automatische Neuberechnung und Umverteilung des Arbeitsplans durch die Engine, sobald sich Voraussetzungen (wie Terminkonflikte, manuelles Verschieben oder Erledigt-Markierungen) ändern.

2.22. UI-Frameworks

Vorgefertigte Software-Gerüste für die Benutzeroberfläche, auf deren Einsatz laut Pflichtenheft zugunsten von reinem CSS-Modules und W3C-Standards explizit verzichtet wird.

2.23. User Experience (Zielwert)

Der angestrebte, optimale Performance-Wert von unter 3 Sekunden für die Generierung eines Arbeitsplans.

2.24. CSS-Modules

Eine Technologie zur lokalen und modularen Einbindung von CSS-Styling, mit der das visuelle Styling und das Fensterlayout der Web-Applikation vollständig eigenständig implementiert werden.

2.25. W3C (Web-Standards)

Gängige und etablierte Web-Standards des World Wide Web Consortiums, nach Version 2.0

denen das visuelle Styling und das Fensterlayout der Applikation zwingend gerichtet sein müssen.

2.26. Identity Management

Die Auslagerung der hochkomplexen und sicherheitskritischen Benutzerauthentifizierung an einen externen Dienst (z. B. Firebase Authentication), um das System durch kryptografische Standards vor Fremdzugriffen zu schützen.

2.27. SDK (Server-seitige Entwicklungs-Schnittstelle)

Eine Programmierschnittstelle zur nahtlosen Anbindung der lokalen Backend-Entwicklungsumgebung an externe Cloud-Dienste (wie Datenbanken und Authentifizierung), die benötigt wird, um die serverseitige Logik abwickeln zu können.

2.28. SDK (Server-seitige Entwicklungs-Schnittstelle)

Eine Programmierschnittstelle zur nahtlosen Anbindung der lokalen Backend-Entwicklungsumgebung an externe Cloud-Dienste (wie Datenbanken und Authentifizierung), die benötigt wird, um die serverseitige Logik abwickeln zu können.