# User Story Whitepaper

Alles Wichtige über

# User Storys und User Story Mapping

auf einen Blick.

t2informatik Wir entwickeln Software.

Eine User Story beschreibt eine Geschichte eines Anwenders. Im Gegensatz zu einer Geschichte, die Sie an einem Lagerfeuer erzählen, sollte eine User Story in kurzen Sätzen und mit einfachen Worten beschrieben werden. So stellen Sie sicher, dass alle an Ihrer Entwicklung beteiligten Kollegen sie auch ohne technischen Hintergrund verstehen. Wie sollte eine User Story beschrieben werden? Was ist der Unterschied von User Storys zu Epics, Features, Use Cases oder Lastenhefte? Und was ist ein User Story Mapping?

## User Story Definition

Der Begriff „User Story“ stammt aus dem Englischen. Das Wort *user* bedeutet Anwender oder Benutzer, *story* bedeutet Geschichte. Im wörtlichen Sinne beschreibt eine User Story also eine Geschichte eines Anwenders. Im agilen Projektmanagement und der Softwareentwicklung ist eine User Story ein Werkzeug, um gewünschte Funktionalitäten eines Systems aus Sicht des Anwenders zu beschreiben. Dabei bietet eine User Story vor allem drei Vorteile:

- sie ist leicht zu verstehen und vermittelt die Wünsche der Anwender
- sie ist schnell erstellt und erleichtert die Schätzung des Aufwands zur Realisierung
- sie lässt sich schrittweise detaillieren und unterstützt so die iterative Entwicklung

Als Konzept geht die User Story auf Extreme Programming (XP) – und nicht wie häufig angenommen auf den Scrum Guide – zurück. Extreme Programming ist ein agiles Entwicklungsmodell, dass zwischen 1995 und 2000 in einem Projekt bei Chrysler von Kent Beck, Ward Cunningham und Ron Jeffries entwickelt wurde.

## Tipps zum Schreiben einer User Story

Im Gegensatz zu einer Geschichte, die Sie an einem Lagerfeuer erzählen, sollte eine User Story in kurzen Sätzen und mit einfachen Worten beschrieben werden. So stellen Sie sicher, dass alle an Ihrer Entwicklung beteiligten Kollegen sie auch ohne technischen Hintergrund verstehen. Es geht vor allem um

- Wer,
- Was und
- Warum,

also wer möchte was von einem System, um welchen Nutzen davon zu haben. Wie die gewünschte Funktionalität später umgesetzt wird, ist für eine User Story unerheblich. Beim Schreiben einer User Story empfiehlt sich folgende Satzstruktur:

**Als (Rolle) möchte ich (Funktionalität), um (Nutzen) zu erreichen.**

Achten Sie bei der Formulierung darauf, sie aus der Sicht des Anwenders, Benutzers oder Kunden zu schreiben und einen wirklichen Nutzen für ihn zu erzeugen.


## Beispiele für eine User Story

Selbst mit der empfohlenen Satzstruktur können User Storys unterschiedlich detailliert sein. Sie können sie relativ grob und nach und nach mit zusätzlichen Details verstehen, so lange, bis sie detailliert genug ist, um sie umzusetzen.

- Als Filmliebhaber möchte ich über neue Filme informiert werden, um zu wissen, welche Filme als nächstes im Kino laufen.
- Als Filmliebhaber möchte ich einmal pro Woche einen Newsletter erhalten, um zu wissen, welche Filme als nächstes im Kino laufen.
- Als Filmliebhaber möchte ich einmal pro Woche per Mail über neue Science Fiction Filme informiert werden, die im ABC Kino laufen, um mir für dieses Kino Tickets online buchen zu können.

## Alternative User Story Satzschablonen

User Storys werden von oder für Anwender oder Kunden geschrieben, um die Funktionalität des zu entwickelnden Systems zu beeinflussen. Neben der am häufigsten genutzten Satzschablone „Als (Rolle) möchte ich (Funktionalität), um (Nutzen) zu erreichen.“ gibt es noch einige Alternativen:

**„Um (Nutzen) als (Rolle) zu erreichen, möchte ich (Funktionalität/Ziel/Wunsch).“**

Diese Formulierung geht auf Chris Matts, einen englischen Programmierer und Experten im Kontext von Agile und Lean Management zurück, der den Wert bzw. Nutzen vor die Funktionalität stellt. Das englische Original lautet: ***„In order to (receive benefit) as a (role), I can (goal/desire).“***

**„Als (Rolle) möchte ich (was) (warum).“**

Diese Formulierung geht auf Rachel Davies, einer Agile Expertin zurück. Das englische Orginal lautet: ***„As (user role), I can (what) so that (why).“***

**„Als (wer) (wann) (wo) möchte ich (was) (warum).“**

Diese Formulierung basiert auf typischen W-Fragen: wer, wann, wo, was, warum. Das englische Original lautet: ***„As (who) (when) (where), I (want) because (why).“***

**„Als (Persona) möchte ich (was) warum.“**

Diese Formulierung geht auf Roman Pichler, einen Produktmanagement-Fachmann, zurück. Das englische Original lautet: ***„As (persona), I want (what) so that (why).“***

Unabhängig von den feinen Unterschieden, ist es empfehlenswert sich für eine Alternative zu entscheiden und die entsprechende Satzschablone beizubehalten. Das erleichtert das Formulieren von User Storys und fördert das Verständnis.

## Unterschiede zwischen User Storys und …

Es gibt eine ganze Reihe von Möglichkeiten, Anforderungen an ein System aus Sicht des Anwenders zu formulieren. Wo liegen die Unterschiede von User Storys zu *Epics und Features*, zu *Use Cases*, zu *Lastenhefte und Pflichtenhefte* oder zu *Technical Storys und Spike Storys*?

### … Epics und Features

Der unterschiedliche Umfang von Funktionalitäten eines Systems und Nutzen aus Sicht eines Anwenders führt häufig zu den Kategorisierungen Epic, Feature und User Story. Die Unterscheidung in diese drei Kategorien ist jedoch nicht standardisiert, d.h. in manchen Unternehmen sind Features umfangreicher als Epics. Wichtig ist daher, dass es im Unternehmen ein einheitliches Verständnis gibt. Die Unterscheidung zwischen Epics, Features und User Storys könnte bspw. wie folgt aussehen:

- Epics beschreiben Funktionalitäten auf höchstem fachlichen Niveau.
- Features verfeinern die Funktionalitäten von Epics und lassen sich für die Release-Planung verwenden. Sie sind aber noch zu umfangreich, um in einer Iteration bzw. einem Sprint umgesetzt zu werden.
- User Storys verfeinern Features und lassen sich in Sprints einplanen und umsetzen.

### … Lastenhefte und Pflichtenhefte

In einem Lastenheft beschreibt ein Kunde seine Wünsche an ein zu entwickelndes System. Dabei landen die Systemanforderungen im Lastenheft, während die Projektanforderungen wie Budgets, Terminpläne, Meilensteine etc. im Projektauftrag oder ggfs. in einem Projekthandbuch festgehalten werden. Ein Pflichtenheft hat eine andere Funktion als ein Lastenheft, denn in ihm werden die Anforderungen

aus dem Lastenheft geklärt und detailliert. Das Pflichtenheft kann klassisch als „System Requirements Specification“ oder im agilen Umfeld mit Epics, Features und User Storys vorliegen. Es kann somit wie ein Product Backlog verstanden werden.

Es kommt vor, dass von Softwareentwicklern geschriebene Pflichtenhefte für Fachabteilungen schwer verständlich sind. Solche Kommunikationsprobleme werden mit agilen Methoden wie Scrum adressiert. In Scrum hat der Product Owner die Aufgabe, aus den Epics, Features und User Storys genauso viele Anforderungen zu generieren, wie im nächsten Sprint realisiert werden können. Im Gegensatz zur Formulierung eines Lastenhefts wird die Vorlaufzeit zur Planung und Realisierung reduziert und die inhaltliche Flexibilität erhöht. Dieser Ansatz ist sehr pragmatisch, stellt aber Anforderungen auf anderen Ebenen wie bspw. Vertragsgestaltung, Aufwandsschätzung und Leistungsabrechnung.

## … Use Cases

Mit einem Use Case beschreiben Sie, wie ein Akteur mit einem zu entwickelnden System interagiert. Use Cases sind sehr beliebt, denn sie ermöglichen ebenso wie User Storys die Erhebung von Anforderungen an ein System. Und beide verfolgen das Ziel, einen Mehrwert für den Anwender zu schaffen. Worin liegen aber die Unterschiede zwischen Use Cases und User Storys?

Ein Use Case ist größer und detaillierter als eine User Story. Er deckt einen Kontext ab und ist damit umfangreicher. Er umfasst somit mehrere User Storys. Und er ist deutlich langlebiger, d.h. er wird über die gesamte Systementwicklung gepflegt, während die User Story mit ihrer Erledigung in einem Sprint praktisch verschwindet.

Use Cases und User Storys ergänzen sich sehr gut. Durch die Kombination beider Methoden lassen sich Anforderungen genauer verstehen. Um Use Cases wie User Storys in eine Sprint-Planung aufzunehmen, muss man sie zu *Use Cases Slices* zerschneiden. Diese Technik nennt sich *Use Case 2.0*.

## … Technical Storys und Spike Storys

Entwicklungsteams nutzen neben User Storys oft auch sogenannte Technical Storys. Mit ihnen werden weder Funktionalitäten noch ein Nutzen eines Anwenders beschrieben, sondern Kriterien, die den technischen Aufwand hinter einer User Story festhalten. So versuchen Entwicklungsteams nicht-funktionale Anforderungen bspw. bezüglich Performance, Skalierung, Sicherheit oder Verfügbarkeit zu definieren. Wichtig ist bei der Erfassung, dass es eine eindeutige Trennung von Technical Storys und User Storys gibt. Durch eine entsprechende Kennzeichnung vermeiden Sie Missverständnisse bei der Priorisierung und Sprint-Planung bspw. im Rahmen eines User Story Mappings.

Eine Spike Story ist eine Art User Story, die verwendet wird, um eine funktionale oder technische Anforderung besser zu verstehen, und so Unsicherheiten zu beseitigen. Sie ist ein Auftrag für eine Analyse, mit der eine Frage beantwortet, Informationen gesammelt oder Projektrisiken adressiert werden. Im Gegensatz zu einer User Story wird sie nicht geschätzt. Das Entwicklungsteam verpflichtet sich, eine bestimmte Zeit zu investieren, um die notwendige Analyse durchzuführen. Das Ergebnis führt zur Überarbeitung der User Story im Sinne einer Verfeinerung, einer Aufteilung in mehrere User Storys oder der Beschreibung einer gänzlich neuen User Story.

## User Story Mapping

Mit steigender Anzahl von Einträgen werden Backlogs leicht unübersichtlich. Hier bietet die Visualisierung mit User Story Maps Abhilfe, denn mit Ihnen lassen sich Zusammenhänge darstellen, die beim Arbeiten mit eindimensionalen Backlogs leicht übersehen werden. Die User Story Map ist eine Art Landkarte, mit der einerseits die Reihenfolge der Verwendung des Systems durch einen Anwender, und andererseits die Zuordnung von User Storys zu den übergeordneten Wünschen der Anwender (also zu den Epics oder Features) visualisiert wird.

Mit dem Story Mapping gelingt es Ihnen leichter, ein Product Backlog zu erstellen. Durch den visuellen, strukturierten Aufbau ist es möglich, ein gemeinsames Verständnis zu schaffen, Lücken im Backlog zu identifizieren und Abhängigkeiten zu erkennen. Zusätzlich kann es auch bei der Aufteilung und Freigabe von Planungsaktivitäten helfen.

Es gibt verschiedene Arten von Mappings mit verschiedenen Formen der Anordnung. So können Sie bspw. Produktvisionen und Ziele, notwendige Maßnahmen, abgeleitete Aufgaben und User Storys visualisieren. Oder Sie stellen Epics, Features, User Storys und Tasks in einer gemeinsamen Visualisierung dar. Auch die Visualisierung von nicht-funktionale Anforderungen, funktionalen Anforderungen und User Storys findet sich in Unternehmen. Wollen Sie Anforderungen aus einer Anwenderperspektive identifizieren, bspw. aus Sicht eines Kunden, eines Mitarbeiters, eines Partners etc., würde sich eine Visualisierung mit Anwender, Zielen, Workflow, Aktionen und User Storys anbieten.

Ob die verwendete Visualisierung der Items top-down, also von oben nach unten, oder von links nach rechts erfolgt, ist für die Anwendung unwesentlich. Wichtig ist, dass Sie für Ihre Entwicklung Abstraktionsebenen finden, mit denen die Beteiligten Anforderungen besser identifizieren und vorhandene Zusammenhänge leichter verstehen können.

## Vorteile beim User Story Mapping

Es gibt verschiedene Vorteile beim Arbeiten mit User Story Maps:

- Items, die in einem haptischen Format erfasst werden, fördern das gemeinsame Verständnis und die Zusammenarbeit im Team.
- Der Backbone zeigt die Highlights der Kundenanforderungen; er ist quasi ein roter Faden und bietet eine gute Orientierung bei der Realisierung der Anforderungen.
- Die Visualisierung vermittelt einen guten Überblick über den Umfang der Anforderungen und Aufgaben.
- Die Priorisierung wird erleichtert, denn Items lassen sich leicht verschieben. Gleichzeitig bleibt der Zusammenhang zu den Kundenanforderungen erhalten.
- Das Mapping ermöglicht die Aufteilung von größeren Items in mehrere kleinere, wobei die Abhängigkeiten untereinander stets nachvollziehbar bleiben.

- Abhängigkeiten lassen sich gut darstellen, so dass besondere Herausforderungen oder Risiken frühzeitig erkannt werden können.
- Das Arbeiten mit User Story Maps macht Spaß und fördert den Dialog im Team.

## User Story Mapping als iterativer Prozess

Das Arbeiten mit Story Maps bietet einen Rahmen für Gespräche zwischen Teams, Teammitgliedern und Stakeholdern. Wie das eigentliche Produkt entwickelt sie sich immer weiter. Sie ist stets eine Momentaufnahme der Gedanken des Teams. Werden in den Gesprächen neue Erkenntnisse gewonnen, Annahmen bestätigt oder widerlegt, ändert sich die Story Map entsprechend.

User Story Mapping beginnt mit der Entscheidung, welches Medium für den Aufbau der Story Map verwendet werden soll. Es kann mit einfachen physischen Ressourcen - wie einer Wand oder einem Whiteboard und Haftnotizen - oder mit einer Vielzahl von Softwaretools durchgeführt werden, die beim Erstellen einer virtuellen Karte für verteilte Teams helfen. Unabhängig vom Medium besteht das Story Mapping aus folgenden Aufgaben:

- Definition der Zielgruppe(n)
- Definition der Ziele der Stakeholder
- Visualisierung des Backbones - also des roten Fadens (des Workflows oder der Geschichte) - der Anwender
- Erstellung und Zuordnung von User Storys zu den Highlight-Anforderungen der Anwender
- Priorisierung der User Storys
- Identifikation von Abhängigkeiten, technischen Anforderungen und Lücken
- Verwendung von Technical Storys und Spike Storys zum Finden von Alternativen und zur Beseitigung von Unklarheiten
- Die Planung von Sprints, Release und Abnahmen

Die Erfahrung zeigt, dass auch zu Beginn getroffene Annahmen regelmäßig überprüft werden sollten. Ziele von Anwendern können sich ebenso ändern, wie die Priorisierung von User Storys oder die technische Möglichkeiten von verwendeten Architekturen.

## Die Zusammenarbeit beim User Story Mapping

User Story Mapping ist ein kollaborativer Ansatz, mit dem funktionsübergreifende Teams Produkte und Systeme entwickeln können. Es macht daher Sinn, dass Vertreter aus den Teams an dem Mapping teilnehmen, die zur Realisierung des Kundenwerts beitragen. Dies können Vertreter aus den Bereichen

- Entwicklung und Architektur
- IT und Operations

- Produktmanagement
- Projektmanageent
- UX Design und Entwurf
- Vertrieb und Marketing
- Finanzen und Recht

sein. Zusätzlich ist die regelmäßige Kommunikation mit Stakeholdern stets zu empfehlen.

## Die Verwendung von Zusatzinformationen

Es gibt Situationen, in denen es Sinn macht, zusätzliche Informationen in einer Story Map zu erfassen. Dies könnten Technical Storys oder Spike Storys sein, Verbindungen zu Personas oder auch einfache "Nice-to-have" User Storys. Richtig ist, was Ihnen in Ihrer Situation hilft. Hier finden Sie einige Möglichkeiten:

- Verwenden Sie verschiedene Farben, um unterschiedliche Ebenen in der Story-Map darzustellen, bspw. orange für Ziele, blau für Epics, grün für Features und Gelb für Storys.
- Verwenden Sie Fäden oder Kreppband, um Zusammenhänge hervorzuheben.
- Nutzen Sie Aufkleber (Sticker) für zusätzliche Informationen, Follow-ups, Notizen oder Fragen.
- Verlinken Sie zusätzliche Informationen (Zeichnungen, Grafiken, Dokumente), um das gemeinsame Verständnis zu erhöhen.
- Bauen Sie Alternativen ein, um technisch robustere oder kostengünstigere Lösungen zu evaluieren.

## Die Identifikation von Schwierigkeiten

User Story Mapping ist ein sehr gutes Werkzeug für agile Teams. Es kann als Werkzeug aber auch Probleme verursachen, wenn es nicht zu Ihrer Unternehmenskultur passt oder Ihre Teams auf eine solche Zusammenarbeit schlecht vorbereitet sind. Hier finden Sie einige Herausforderungen, auf die Sie achten sollten:

- Wer ist Ihr Anwender und welche Ziele verfolgt er?

  Wenn Sie nicht wissen, wer die Anwender Ihrer Lösung sind, ist es unmöglich herauszufinden, wie er das Produkt erleben wird.

- Was ist die zu lösende Aufgabenstellung?

  Wenn Sie nicht wissen, welches Problem Ihr Produkt für Kunden löst, kann die gesamte User Story Zuordnung leicht scheitern. User Storys auf ein falsches Kundenziel auszurichten, kann zu einer Verschwendung von Zeit und Ressourcen führen - nicht nur im Mapping, sondern auch später bei der Realisierung in den Sprints und Releases, die darauf basieren.

- Wie erfolgt die Aktualisierung Ihrer Story Map?

  Physische Story Maps, die aus Haftnotizen an einer Wand oder einem Whiteboard bestehen, sind schwer zu aktualisieren. Die Notizen fallen herunter, Whiteboards werden gereinigt und die Arbeit geht verloren. Es passiert auch, dass Releases ohne Updaten der Story Map ausgeliefert werden, d.h. Planung und Realität laufen auseinander.

- Wie gewährleisten Sie den Zugang zur Story Map?

  User Story Maps, die an einem einzigen physischen Standort erstellt wurden, stehen häufig Teams an anderen Standorten nicht zur Verfügung. Und auch bei der Verwendung an einem Standort mit mehreren Teams muss der Zugang zu den Informationen stets gewährleistet sein.

- Wie vermeiden Sie Nacharbeiten und Redundanzen?

  Erkenntnisse aus dem Arbeiten mit einer User Story Map müssen oft anschließend in einem Product Backlog neu erstellt oder überarbeitet werden. So kann sich das Gefühl einstellen, dass dieselbe Arbeit zweimal durchgeführt wird.

## Aufwand einer User Story schätzen

Wie schätzen Sie den Aufwand einer User Story? Die Antwortet lautet: in Personentagen oder mit Story Points. Beim Arbeiten mit Story Points bestimmt das Entwicklungsteam ein Kriterium oder eine Verknüpfung von mehreren Kriterien, um die Größe der User Story festzulegen. Ein solches Kriterium könnte Komplexität sein, bspw. in Bezug auf die Verwendung von verschiedenen Schichten des Architekturmodells. Es geht also bei Story Points nicht um die Zeit, die zur Umsetzung der User Story benötigt wird, sondern um die strukturellen Eigenschaften einer User Story. Natürlich kann ein erfahrenes Entwicklungsteam im Vergleich zu einem weniger erfahrenen Team größere User Storys in einem Sprint umsetzen, doch die Größe der Story bleibt davon unberührt. In anderen Worten: die Eigenschaften einer User Story hängen nicht von den Fähigkeiten des Teams ab.

In Scrum drückt die sogenannte *Velocity* aus, wie viele Story Points ein Team pro Iteration umsetzt, so dass auch beim Arbeiten mit Story Points Aussagen getroffen werden können, in welcher Iteration ein Feature geliefert wird. In der Praxis können Merkmale einer User Story häufig ohne detaillierte Tätigkeitenanalyse identifiziert werden. Dies ist ein wesentlicher Unterschied zu der Aufwandsschätzung in Personentagen, bei der alle notwendigen Tätigkeiten identifiziert und summiert werden. Oft wird eine User Story mit mittlerer Größe als Referenz ausgewählt und die Schätzung im Vergleich zu dieser Referenz durchgeführt. Als Wertebereich dient eine Fibonacci-Reihe bis 13, ergänzt mit 20, 40 und 100, wobei 1 einer Aufgabe mit niedriger Komplexität und 100 einer noch nicht einschätzbare Aufgabe entspricht.

## Das INVEST-Prinzip

Wie stellen Sie fest, ob Sie eine qualitativ gute User Story definiert haben? William Wake bietet mit seinem INVEST-Prinzip Hilfe bei der Formulierung von User Storys. INVEST ist ein Akronym:

- Indepentent: Die User Story steht für sich und ist unabhängig von anderen Storys.
- Negotiable: Der Inhalt eine User Story ist verhandelbar und wird nach und nach detaillierter beschrieben, bis sie sich umsetzen lässt.
- Valuable: Die User Story ist wertvoll und bietet dem Anwender oder Kunden einen Mehrwert bzw. Nutzen.
- Estimable: Der Aufwand der User Story muss sich durch die Entwickler schätzen lassen.
- Small: Die User Story ist so klein, dass sie innerhalb einer Iteration bzw. eines Sprints realisierbar ist.
- Testable: Die User Story verfügt über Akzeptanzkriterien und ist prüfbar.

## DEEP im Product Backlog

User Storys werden in Backlogs verwaltet. Hier kommt das Akronym DEEP zum Tragen, das Roman Pichler in seinem Buch „Agile Product Management with Scrum“ erläutert:

- **D**etailed Appropriately (angemessen detailliert): User Storys, die demnächst umgesetzt werden sollen, müssen detaillierter ausgearbeitet sein als jene, die erst zu einem späteren Zeitpunkt umgesetzt werden sollen.
- Estimated (geschätzt): Backlog Items werden geschätzt, wobei Items mit höherer Priorität zunächst detaillierter geschätzt werden als Items mit niedrigerer Priorität.
- **E**mergent (enstehend): Im Laufe von Entwicklungen werden neue Informationen und Erkenntnisse gewonnen. User Storys werden folglich im Produkt Backlog hinzugefügt, entfernt oder neu geordnet.
- **P**rioritized (priorisiert): Alle Items werden im Backlog nach Priorität geordnet. Ziel ist es, die wichtigsten zuerst zu implementieren.

DEEP gilt als nützliches Konzept, das die kontinuierliche Arbeit mit Backlog Items bzw. User Storys betont.

## Akzeptanzkriterien einer User Story

Wann wissen Sie, ob eine User Story vollständig implementiert wurde? Hier helfen Akzeptanzkriterien. Mit Akzeptanzkriterien legen Sie stichpunktartig Ergebnisse fest, die durch die User Story erfüllt werden müssen.

Akzeptanzkriterien gelten als Bindeglied zwischen User Storys und Testfällen. Eine Möglichkeit, Akzeptanzkriterien zu definieren, ist das Hinterfragen von Schlüsselwörtern, also Verben, Adjektiven und Substantiven. Beispiel:

„Als Kinobesucher möchte ich meine gekauften Tickets in meinem Profil speichern, damit ich nachvollziehen kann, welche Filme ich gesehen habe."

- Wer speichert was, wann, wo?
- Was geschieht bei der Speicherung eines neuen Tickets mit bereits gespeicherten Informationen?
- Wie viele Tickets sollen gespeichert werden können?

Mit solchen W-Fragen finden Sie leicht Akzeptanzkriterien. Als Checkliste sind sie eine gute Basis für die Entwicklung von umfangreichen Testfällen.

**Über t2informatik**

**Wir entwickeln Software.**

Der Erfolg von IT-Projekten ist für die meisten Unternehmen und Organisationen sehr wichtig. Dafür entwickeln wir mit großer Leidenschaft und viel Know-how Software. Individuell nach den Anforderungen unserer Kunden. Wir sind

- Softwareentwickler und Softwarearchitekten für .NET, Java und Web-Technologien,
- Projektmanager, Anforderungsanalytiker und Testmanager,
- Frauen und Männer, Singles, Eheleute und Eltern.

Wir tanzen, spielen Fußball oder fahren Motorrad. Wir sind Experten in der Softwareentwicklung und wir lernen jeden Tag dazu. Wir glauben an Vielfalt und Gleichberechtigung. Wir arbeiten in Teams in unseren Büros in Berlin, von zuhause oder bei unseren großartigen Kunden im deutschsprachigen Raum. Gerne demnächst auch bei Ihnen.

**Was dürfen wir für Sie tun?**

**Sagen Sie es uns einfach unter** https://t2informatik.de/kontakt/anfrage-stellen/.

Wir freuen uns darauf. Bei Interesse empfehlen wir Ihnen auch gerne eine Expertin oder einen Experten aus unserem Netzwerk.

PS: Wir suchen übrigens auch neue Mitarbeiter. Unter https://t2informatik.de/jobs/ finden Sie weitere Informationen.