

# Tesler's Law

## Einführung

Tesler's Law beschreibt das Prinzip, dass jede Aufgabe eine gewisse unvermeidbare Komplexität besitzt. Diese Komplexität kann nicht vollständig entfernt werden. Sie kann nur unterschiedlich verteilt werden, entweder auf das System oder auf den User. Für die Gestaltung digitaler Produkte bedeutet das, dass gute UX nicht darin besteht, Komplexität verschwinden zu lassen, sondern sie so zu organisieren, dass die Nutzung möglichst verständlich und handhabbar bleibt.

Dieses Gesetz ist besonders wichtig, weil es vor einer typischen Fehlannahme schützt: Nicht jede schwierige Aufgabe lässt sich durch Design in etwas vollkommen Einfaches verwandeln. Manche Prozesse sind komplex, weil die Sache selbst komplex ist. Die eigentliche Gestaltungsaufgabe besteht dann darin, zu entscheiden, welcher Teil dieser Komplexität im Hintergrund vom System übernommen werden sollte und welcher Teil vom User sinnvollerweise bewusst bearbeitet werden muss.

## Warum ist das relevant?

Viele digitale Produkte unterstützen Aufgaben, die inhaltlich anspruchsvoll sind. Dazu gehören zum Beispiel Buchungen, Konfigurationen, Vertragsabschlüsse, Datenauswertungen, medizinische Angaben oder Verwaltungsvorgänge. In solchen Fällen entsteht leicht der Wunsch, die Oberfläche müsse alles radikal vereinfachen. Genau hier setzt Tesler's Law an. Es macht deutlich, dass Vereinfachung Grenzen hat.

Für User ist dieses Prinzip deshalb relevant, weil schlechte UX oft dadurch entsteht, dass unvermeidbare Komplexität falsch verteilt wird. Entweder muss der User zu viel selbst verstehen, merken und entscheiden, oder das System verbirgt so viel, dass wichtige Zusammenhänge unklar werden. Gute UX sucht hier ein Gleichgewicht. Sie nimmt Belastung ab, ohne den Inhalt unverständlich zu machen.

## Genaues Prinzip

Tesler's Law besagt, dass Komplexität in einer Aufgabe immer irgendwo bestehen bleibt. Wenn ein Interface oberflächlich sehr einfach wirkt, heißt das nicht automatisch, dass die zugrunde liegende Komplexität verschwunden ist. Häufig wurde sie lediglich in den Hintergrund verlagert, etwa in Systemlogik, Automatisierung, Voreinstellungen oder intelligente Vorschläge. Das kann sehr sinnvoll sein, solange der User dadurch tatsächlich entlastet wird und nicht die Kontrolle verliert.

Für die Praxis bedeutet das, dass Designer nicht nur fragen sollten, wie etwas einfacher aussieht, sondern wie Komplexität sinnvoll verteilt wird. Manche Entscheidungen sollte das System automatisch übernehmen, wenn sie standardisierbar und risikoarm sind. Andere Entscheidungen sollten bewusst beim User bleiben, wenn sie Kontextwissen, Verantwortung oder individuelle Präferenzen erfordern. Gute UX besteht also nicht darin, jede Komplexität zu verstecken, sondern darin, sie passend zu verlagern.

## Ausführliche Anwendungsbeispiele

### Beispiel 1: Buchungs- oder Checkout-Prozesse

Ein gutes Beispiel ist ein Buchungsprozess für Reisen, Veranstaltungen oder Dienstleistungen. Die Aufgabe

selbst ist häufig komplex, weil Termine, Preise, Verfügbarkeiten, Personenangaben, Zahlungsarten und Bedingungen berücksichtigt werden müssen. Diese inhaltliche Komplexität verschwindet nicht einfach dadurch, dass die Oberfläche minimalistisch gestaltet ist.

Eine schlechte Lösung wäre, alle Entscheidungen gleichzeitig dem User zu überlassen und alle Details ungefiltert auf einer einzigen Seite zu zeigen. Ebenso problematisch wäre aber eine zu starke Vereinfachung, bei der das System wichtige Einschränkungen oder Konsequenzen verbirgt. Eine gute Lösung übernimmt dort Komplexität, wo das sinnvoll ist. Das System kann etwa Standardoptionen vorauswählen, Zwischenschritte logisch strukturieren, Abhängigkeiten automatisch berechnen und nur relevante Informationen im jeweiligen Moment anzeigen. Gleichzeitig muss der User wichtige Entscheidungen nachvollziehen können, etwa bei Zusatzkosten, Stornierungsbedingungen oder Terminwahl. Das System reduziert also nicht die Komplexität der Aufgabe selbst, sondern organisiert sie.

## **Beispiel 2: Formulare mit komplexen Anforderungen**

Tesler's Law zeigt sich auch sehr klar in umfangreicheren Formularen, etwa bei Versicherungen, behördlichen Anträgen oder professionellen Software-Systemen. Die Inhalte solcher Formulare sind oft nicht deshalb schwierig, weil sie schlecht gestaltet sind, sondern weil sie rechtliche, organisatorische oder fachliche Komplexität abbilden. Diese Komplexität lässt sich nicht vollständig entfernen.

Eine gute UX wird hier nicht versuchen, den gesamten Vorgang künstlich harmlos erscheinen zu lassen. Stattdessen unterstützt sie den User durch klare Struktur, kontextbezogene Hilfen, sinnvolle Vorausfüllungen und verständliche Rückmeldungen. Das System kann Plausibilitätsprüfungen übernehmen, irrelevante Felder ausblenden und Zusammenhänge sichtbar machen. Der User muss sich dennoch mit den inhaltlich wichtigen Punkten auseinandersetzen. Genau darin liegt die Logik von Tesler's Law: Nicht alles kann weg vereinfacht werden, aber vieles kann besser verteilt werden.

## **Beispiel 3: Erweiterte Einstellungen in Software**

Ein weiteres typisches Beispiel sind Einstellungsbereiche in komplexeren Anwendungen. Viele Produkte haben sowohl einfache Standardnutzung als auch fortgeschrittene Konfigurationsmöglichkeiten. Würde man alle Optionen immer sofort sichtbar machen, entstünde für viele User eine unnötig hohe Belastung. Würde man dagegen alle erweiterten Möglichkeiten verstecken oder entfernen, könnten erfahrene Nutzer wichtige Kontrolle verlieren.

Eine gute Gestaltung löst dieses Problem oft durch Ebenen der Komplexität. Standardoptionen werden klar und einfach zugänglich gemacht, während erweiterte Einstellungen bewusst nachgelagert sind. Auf diese Weise übernimmt das System einen Teil der Komplexität für typische Nutzungssituationen, ohne sie vollständig zu eliminieren. User mit mehr Erfahrung oder spezielleren Anforderungen können tiefer einsteigen, wenn es nötig ist. Das ist ein sehr direktes Beispiel dafür, wie Komplexität sinnvoll verteilt werden kann.

## **Beispiel 4: Automatisierung und Kontrolle**

Auch Automatisierung lässt sich gut mit Tesler's Law erklären. Viele digitale Produkte versuchen, Entscheidungen automatisch im Hintergrund zu treffen, etwa bei Sortierungen, Empfehlungen, Vervollständigungen oder Standardwerten. Das kann die Nutzung deutlich erleichtern, weil der User weniger manuell einstellen oder entscheiden muss.

Problematisch wird es dann, wenn die Automatisierung die zugrunde liegende Komplexität zwar verbirgt, aber nicht mehr nachvollziehbar macht. Wenn ein System etwas automatisch auswählt, ohne dass klar ist warum oder wie es geändert werden kann, entsteht Unsicherheit. Gute UX entlastet also durch Automatisierung, ohne

den User zu entmündigen. Die Komplexität wird ins System verlagert, bleibt aber transparent genug, um Vertrauen und Korrekturmöglichkeiten zu erhalten.

## Was für die Nutzung besonders wichtig ist

Aus Sicht des Users ist ein gutes Produkt nicht dadurch gekennzeichnet, dass es jede Schwierigkeit unsichtbar macht. Entscheidend ist vielmehr, dass notwendige Komplexität nicht unnötig belastend wird. User profitieren davon, wenn das System Routine, Struktur und technische Logik übernimmt, während wichtige Entscheidungen klar, verständlich und kontrollierbar bleiben.

Tesler's Law ist deshalb besonders wertvoll bei komplexeren Produkten und Prozessen. Es erinnert daran, dass gute UX nicht nur reduziert, sondern verteilt. Eine Oberfläche kann sehr einfach wirken und trotzdem schlecht sein, wenn sie wichtige Zusammenhänge verbirgt. Umgekehrt kann ein komplexes Produkt gut nutzbar sein, wenn die Komplexität sinnvoll organisiert und an den richtigen Stellen getragen wird.

## Typische Missverständnisse

Ein häufiges Missverständnis besteht darin, Tesler's Law so zu verstehen, als sei Vereinfachung grundsätzlich unmöglich. Das ist nicht gemeint. Natürlich kann und sollte UX viele Dinge erleichtern. Das Gesetz sagt nur, dass die Komplexität einer Aufgabe nicht vollständig verschwindet. Sie wird höchstens anders aufgeteilt.

Ein weiteres Missverständnis ist die Annahme, dass möglichst viel Komplexität immer vom System übernommen werden sollte. Auch das stimmt nicht. Manche Entscheidungen müssen bewusst beim User bleiben, weil sie individuell, folgenreich oder kontextabhängig sind. Gute UX nimmt daher nicht einfach alles ab, sondern verteilt Verantwortung und Aufwand so, dass die Nutzung verständlich, effizient und vertrauenswürdig bleibt.

From:

<https://wiki.ct-lab.info/> - Creative Technologies Lab | dokuWiki

Permanent link:

<https://wiki.ct-lab.info/doku.php/extras:wissikon:media-design:ten-ux-laws:teslers-law?rev=1776759269>

Last update: 2026/04/21 08:14

