



B2 Was ist ein „Bug“ ?

Der folgende Text ist ein Auszug aus der deutschen [Wikipedia zum Thema „Programmfehler“](#). Er soll als Hilfestellung für die Bearbeitung des Arbeitsblatts “Fehler: Bugs und Diskriminierung” dienen.

! Wikipedia ist eine freie Online-Enzyklopädie, an der jede Person mitarbeiten kann. Die Inhalte haben deshalb keinen Anspruch auf Richtigkeit und Vollständigkeit. Sie können zur ersten Orientierung genutzt werden, sollten aber nicht übernommen oder zitiert werden. Es kann auch wichtig sein, zu hinterfragen, wer die Artikel schreibt und welche Zielgruppe diese Person im Kopf hat.

The screenshot shows the Wikipedia article for 'Programmfehler'. On the left is the Wikipedia sidebar with navigation links. The main content area has a title 'Programmfehler' and a 'Diskussion' tab. The text explains that a program error (Bug) is a deviation from the specification, often caused by incorrect implementation or flawed testing environments. It also mentions tools like debuggers and formal verification for finding and preventing errors.

WIKIPEDIA
Die freie Enzyklopädie

Hauptseite
Themenportale
Zufälliger Artikel

Mitmachen

Artikel verbessern
Neuen Artikel anlegen
Autorenportal
Hilfe
Letzte Änderungen
Kontakt
Spenden

Werkzeuge

Links auf diese Seite
Änderungen an verlinkten Seiten
Spezialseiten
Permanenter Link
Seiteninformationen
Artikel zitieren
Wikidata-Datenobjekt

Drucken/exportieren
Buch erstellen

Artikel **Diskussion**

Programmfehler

Ein **Programmfehler** oder **Softwarefehler** oder **Software-Anomalie**, häufig auch **Bug** (*englisch*) genannt, bezeichnet im Allgemeinen ein Fehlverhalten von **Computerprogrammen**. Dies tritt auf, wenn der **Programmierer** eine bestimmte Festlegung der **Spezifikation** nicht oder falsch **umgesetzt** hat, oder wenn die **Laufzeitumgebung** fehlerhaft bzw. anders als erwartet arbeitet. Weiterhin können auch Unvollständigkeit, Ungenauigkeit oder Mehrdeutigkeiten in der Spezifikation des Programms zu „Fehlern“ führen.

Zur möglichst vollständigen Erkennung und Behebung von Programmfehlern wird üblicherweise in den Prozessen der **Softwareentwicklung**, d. h. vor dem tatsächlichen, „produktiven“ Einsatz von Software, die **Projektphase Softwaretest** durchlaufen, wobei eine Validierung durchgeführt wird. Dabei auftretende Fehler sind üblich und sie zu finden ist Ziel des Testens,^[1] während Fehler im laufenden Betrieb je nach Fehlerwirkung u. U. kritische Anomalien/Störungen darstellen. In der Praxis treten **Computerprogramme** ohne Programmfehler selten auf. Als Qualitätsmerkmal für Programme kennt man u. a. die **Fehlerdichte**. Sie bezeichnet die Anzahl an Fehlern pro 1.000 Zeilen Code (**Kilo Source Lines of Code**) bzw. je **Function Point**.

Als spezielle Instrumente zur Suche nach den Ursachen für Fehler in Programmen sind sogenannte **Debugger** hilfreich, mit denen ein Programm Schritt für Schritt ausgeführt und kontrolliert werden kann. Bei besonders kritischer Software (z. B. Flugzeugsteuerung) wird mitunter eine (aufwendige) **formale Verifikation** durchgeführt.

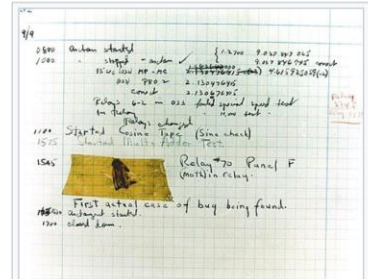
Zur Erfassung und Dokumentation werden sogenannte **Bugtracker** (wie **Bugzilla** oder **Mantis**) eingesetzt. Diese nehmen sowohl **Fehlerberichte** als auch Verbesserungsvorschläge und Wünsche (sog. **Feature-Requests**) der Nutzer oder allgemeine Vorgänge auf. Siehe auch **Fehlermanagement**.

Abbildung 1 Auszüge aus <https://de.wikipedia.org/wiki/Programmfehler>



„Bug“ als Synonym für Programmfehler [Bearbeiten | Quelltext bearbeiten]

Das Wort *bug* bedeutet im Englischen „Schnabelkerf“, **Wanze**“ und umgangssprachlich „landlebender Gliederfüßer“ oder „(insektenartiges) Ungeziefer“.^[3] Im Jargon amerikanischer Ingenieure ist seit dem späten 19. Jahrhundert die Bedeutung „Fehlfunktion“ oder auch „Konstruktionsfehler“ bezeugt; diesem Wortgebrauch liegt die (scherzhafte) Vorstellung zugrunde, dass sich kleines Krabbeltier am Getriebe, der Leitung usw. zu schaffen macht. [...]



Logbuch-Seite des Mark II Aiken Relay Calculator mit dem ersten *bug* (1947)

Arten von Programmfehlern [Bearbeiten | Quelltext bearbeiten]

In der **Softwaretechnik** (siehe auch^[6]) wird zwischen folgenden Typen von Fehlern in Programmen unterschieden:

- **Lexikalische Fehler** sind nicht interpretierbare Zeichenketten, also undefinierte Bezeichner (Variablen, Funktionen, **Literale**...)
- **Syntaxfehler** sind Verstöße gegen die grammatischen Regeln der benutzten **Programmiersprache**, zum Beispiel die falsche Verwendung reservierter Symbole (z. B. fehlende Klammern), Typkonflikte, falsche Anzahl Parameter.

Lexikalische und Syntaxfehler verhindern in der Regel die **Kompilierung** des fehlerhaften Programms und werden daher frühzeitig erkannt. Bei Programmiersprachen, die sequentiell **interpretiert** werden, bricht das Programm üblicherweise erst an der syntaktisch/lexikalisch fehlerhaften Stelle ab.

- **Semantische Fehler** sind Fehler, in denen eine programmierte Anweisung zwar syntaktisch fehlerfrei, aber inhaltlich trotzdem fehlerhaft ist, zum Beispiel Verwechslung des Befehlscodes, syntaktisch nicht erkennbare falsche Parameterreihenfolge.
- **Logische Fehler** bestehen in einem im Detail falschen Problemlösungsansatz, beispielsweise auf Grund eines **Fehlschlusses**, einer falsch interpretierten **Spezifikation** oder einfach eines Versehens oder Schreibfehlers. Beispiele: plus statt minus, kleiner statt kleiner/gleich usw. Die Toleranz gegenüber solchen Fehlern und die diese einschränken sollende **Attributgrammatik** von Programmiersprachen, wie etwa bei der **Zuweisungskompatibilität** von **Datentypen**, sind je nach verwendeter **Programmiersprache** sehr unterschiedlich ausgeprägt und können schwierig zu überschauende **Sicherheitslücken** und **Programmabstürze** verursachen.
- **Designfehler** sind Fehler im Grundkonzept, entweder bei der Definition der **Anforderungen** an die **Software**, oder bei der Entwicklung des Softwaredesigns, auf dessen Grundlage das Programm entwickelt wird. Fehler bei der **Anforderungsdefinition** beruhen oft auf mangelnder Kenntnis des Fachgebietes, für das die Software geschrieben wird oder auf Missverständnissen zwischen Nutzern und Entwicklern. [...]
- Fehler im **Bedienkonzept**. Das Programm verhält sich anders als es einzelne oder viele Anwender erwarten, obwohl es technisch an sich fehlerfrei arbeitet.

Abbildung 2 Auszüge aus <https://de.wikipedia.org/wiki/Programmfehler>



Sonstige Fehlerbegriffe

- **Laufzeitfehler:** Während die vorgenannten Fehler ein tatsächlich fehlerhaftes Programm bedeuten, das entweder nicht ausführbar ist oder fehlerhafte Ergebnisse liefert, kann auch ein „korrektes“ Programm bei seiner Ausführung zu Fehlern führen. Laufzeitfehler sind alle Arten von Fehlern, die auftreten, während das Programm abgearbeitet wird. Je nach Situation kann die Ursache beispielsweise eine unpassenden Programmumgebung sein (z. B. eine falsche **Betriebssystem**-Version, falsche **Parameter** beim Aufruf des Programms (auch als **Unterprogramm**), falsche Eingabedaten etc.)
 - Fehler im **Compiler, der Laufzeitumgebung oder sonstigen Bibliotheken**. Solche Fehler sind meist besonders schwer nachzuvollziehen, da das Verhalten des Programms in solchen Fällen nicht seiner Semantik entspricht. Insbesondere von Compiler und Laufzeitumgebung wird daher besondere Zuverlässigkeit erwartet.
 - Ein **Regressionsbug** (*Regression* bedeutet „Rückschritt“) ist ein Fehler, der erst in einer späteren Programmversion auftaucht. Dies sind häufig unerkannte Nebeneffekte von Fehlerbehebungen oder Programmänderungen an anderer Stelle.
 - Fehler als Folge physikalischer **Betriebsbedingungen**. Verschiedenste Begebenheiten wie elektromagnetische Felder, Strahlen, Temperaturschwankungen, Erschütterungen usw. können auch bei sonst einwandfrei konfigurierten und innerhalb der Spezifikationen
- Bei manchen Projekten wird nicht der Begriff Bug verwendet, sondern man spricht zum Beispiel von Metabugs, bei denen ein Bug ein Element einer Aufgabenliste darstellt. Bei einigen Projekten spricht man stattdessen auch von „Issues“ (Angelegenheiten), da sich dieser Ausdruck nicht auf Programmfehler beschränkt.
- Konkrete Beispiele von Fehlern mit medial besonderer Wirkung finden sich in der [Liste von Programmfehlerbeispielen](#).

Abbildung 3 Auszüge aus <https://de.wikipedia.org/wiki/Programmfehler>

Folgen von Programmfehlern [[Bearbeiten](#) | [Quelltext bearbeiten](#)]

Die Folgen von Programmfehlern können hochgradig unterschiedlich sein und sich in vielfältiger Weise zeigen. Werden Fehler im Rahmen der Entwicklungsprozesse entdeckt, so beschränken sich die Fehlerfolgen außerdem auf die Überarbeitung der Software (Codekorrekturen, Konzeptüberarbeitung, Dokumentation ...) – je nach Situation mit mehr oder weniger großen Auswirkung auf das Projektbudget und die Projektdauer. Dagegen wirken erst im Produktivbetrieb erkannte Fehler nicht selten ungleich kritischer, zum Beispiel können sie Prozess-Störungen oder Produktionsstillstand bewirken, Imageschäden hervorrufen, den Verlust von Kunden und Märkten verursachen, Regresspflichten auslösen oder gar das Unternehmen in Existenzgefahr bringen. Fehler in technischen Anwendungen können im schlimmsten Fall zu Katastrophen führen.

Konkrete Beispiele für Programmfehler und deren Folgen finden sich in der [Liste von Programmfehlerbeispielen](#).

Abbildung 4 aus <https://de.wikipedia.org/wiki/Programmfehler>