

### **Bug tracking Software:**

- Bugtracking bezeichnet den Prozess des Identifizierens, Überwachen und protokollieren sowie Priorisieren und beheben von Bugs
- Bugs können mittels Bug tracking System über einen langen Zeitraum verfolgt werden
- Bug tracking Systeme kommen als Webbasierte oder als Desktopbasierte Anwendungen

### **Probleme die entstehen ohne Bugtracking:**

- Mangelnde Kommunikation
- Fehlen von Echtzeit-Updates
- Mangel an fließenden Prioritäten
- Fehlen eines zentralen Speichers
- Fehlende Einsichten

### **Vorteile von Bugtracking:**

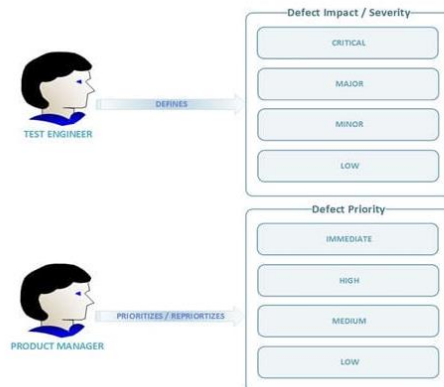
- Bieten mehr Transparenz und eine bessere Kommunikation innerhalb des Teams
- Es ermöglicht Entwicklern eine einfachere und effizientere Zusammenarbeit bei Projekten
- Es reduziert die Anzahl der Fehler in der Software und stellt sicher, dass alle Probleme schnell erkannt und behoben werden
- Es hilft dabei, den Überblick über die zugewiesenen Aufgaben zu behalten und den Fortschritt ihrer Projekte zu überwachen
- Es hilft den Entwicklern, genauere Zeitpläne zu erstellen und die Fertigstellung wichtiger Meilensteine zu verfolgen

### **Nachteile von Bugtracking:**

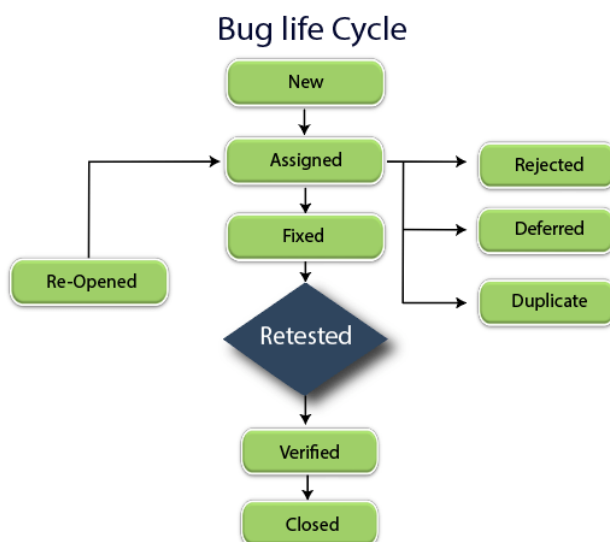
- Die Bugs müssen gut und detailliert dokumentiert werden.
- Kosten für zusätzliches Personal falls benötigt
- Es kann schwierig sein, eine gute Software zur Fehlerverfolgung zu finden, die den Anforderungen Ihres gesamten Entwicklungsteams gerecht wird
- Es kann schwierig sein, die Fehlerverfolgungssoftware in Ihre bestehenden Systeme oder Arbeitsabläufe zu integrieren
- Alle Teammitglieder müssen die Software verstehen und anwenden können

## Wie funktioniert Bugtracking:

Wenn ein Bug vom Testteam entdeckt wird, wird er nach Priorität und Schweregrad eingeordnet



In den meisten Fällen werden die Zustände und Schweregrade in einer Fehlerdatenbank überwacht. Eine gute Tracking-Plattform ist auch mit größeren Softwareentwicklungs- und Managementsystemen verknüpft: So lassen sich der Fehlerstatus und die potenziellen Auswirkungen auf die Gesamtproduktion und den Zeitplan besser beurteilen.



Der Bug-life Cycle (Bsp. in Jira) veranschaulicht die verschiedenen Zustände die ein Bug in einer Bug tracking Software durchläuft. Diese können je nach System etwas variieren haben aber alle den gleichen Ansatz.

# GitHub Issues

## Was ist GitHub Issues

GitHub Issues wird verwendet um Ideen, Feedback, Aufgaben oder Fehler für die Arbeit auf GitHub nachzuverfolgen, wo eben die Entwicklung passiert

Issues können auf verschiedene Arten erstellt werden, sodass du die bequemste Methode für deinen Workflow auswählen kannst.

Um über die neuesten Kommentare in einem Issue auf dem Laufenden zu bleiben, kannst du ein Issue abonnieren, um Benachrichtigungen zu den neuesten Kommentaren zu erhalten

Du kannst die Projektmitarbeiter\*innen mit „@“ erwähnen (@mention), die Zugriff auf dein Repository in einem Issue haben, um deren Aufmerksamkeit auf diesen Kommentar zu lenken.

## Repository erstellen

Man kann ein neues Repository in seinem persönlichen Konto oder in jeder Organisation erstellen, für die du ausreichend Berechtigungen hast.

Es gibt auch gewisse Vorlagen, die eine Struktur im Repository vorgeben.

## Issues erstellen

Auf der Hauptseite des Repositorys findet man „New Issue“

Es gibt eben verschiedene Arten um ein Issue zu erstellen (z.Bsp. Code oder Kommentar)

Auch hier gibt es Vorlagen, die einen durchgehenden Standard für jedes Issue sicherstellen

## Standardkennzeichnung von einem Issue

Füge eine Kennzeichnung hinzu, um dein Problem zu kategorisieren.

Bug	Gibt ein unerwartetes Problem an
Documentation	Kennzeichnet Verbesserungen oder Ergänzungen der Dokumentation
Enhancement	Kennzeichnet neue Funktionsanfragen
Good first issue	Kennzeichnet einen geeigneten Issue für erstmalig Mitwirkende

## **Pull request erstellen**

Man kann einen Pull Request mit einem Issue verknüpfen, um anzuzeigen, dass ein Fix (also die Behebung von einem Problem) in Bearbeitung ist und dass das Issue automatisch geschlossen werden soll, wenn der Pull Request gemergt wird. (praxisbeispiel)

## **Open Source Project**

Sample Gatsby

Static Site Generator für statische Websites

(Statische WS = sind zwar interaktiv, aber limitiert was die Daten angeht zb keine Chatfunktion)

How to Fork? Also wie man es zu „seinem“ project macht

How to Find? Website für open source projects

<https://github.com/gatsbyjs/gatsby>

<https://goodfirstissue.dev/language/javascript>

# **Bugzilla – Bug tracking tool**

## **Entstehung von Bugzilla:**

- Ursprünglich von Terry Weissman bei Netscape in TLC geschrieben
- Ersatz für rudimentäre Bug-Tracking-Datenbank
- 1998 nach Perl optimiert (Bugzilla 2.0)
- Seit 2001 Leiter Dave Miller
- aktuelle Version 5.0.6

## **Was ist notwendig um Bugzilla zu installieren:**

- Perl
- Oracle / MySQL oder PostgreSQL
- Webserver
- Bugzilla
- Perl Module
- Mail Transfer Agent

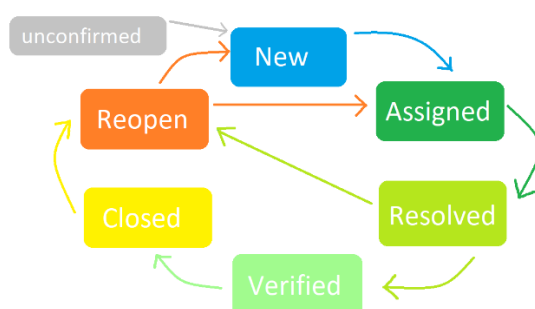
## **Vorteile von Bugzilla:**

- Open source software
- Free software
- Betriebssystem: plattformunabhängig
- Erweiterte Suchfunktionen
- Bug Liste in unterschiedlichen Formaten exportierbar
- Automatische Erkennung von Bug Duplikaten

## **Nachteil von Bugzilla:**

- Sieht veraltet aus
- Nicht cloud-basierend

## **Life Cycle of a bug in Bugzilla**



# Bugzilla

## Documentation:

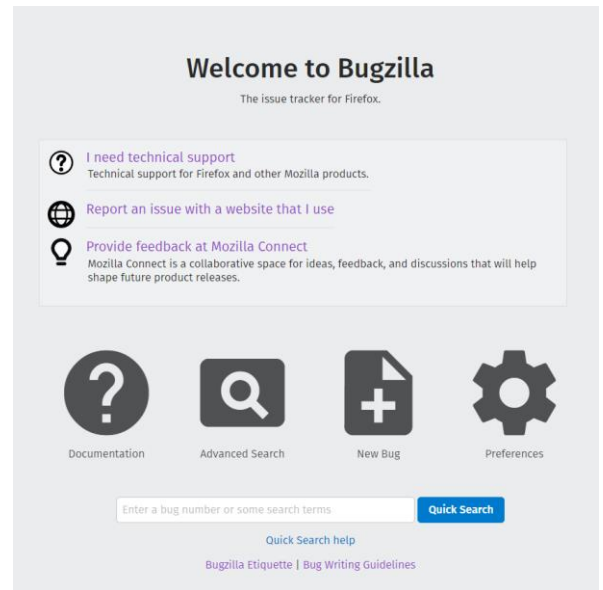
Guide von Bugzilla mit allen Informationen und Erklärungen wie zum Beispiel: Installation, Administration, Sicherheit, Verwendung, Anpassungen, usw

## Advanced Search:

4 Arten von Suchfunktionen: Instant Search, Simple Search, Advanced Search, Google Search

## Preferences:

Standardeinstellungen an die eigenen Bedürfnisse anpassen und ändern



## New Bug:

- **Product:** Bugs sind in Produkte und Komponenten unterteilt; dies dient zur Klassifizierung
- **Component:** Kategorie der zweiten Ebene und dient zur Eingrenzung der Liste
- **Summary:** kurze und bündige Fehlerzusammenfassung in einem Satz
- **defect (Type):** Bug führt zum Absturz des Programms; Programm bleibt hängen; Sicherheitslücken bestehen
- **enhancement (Type):** Verbesserungsvorschläge, aber keine technischen Änderungen
- **task (Type):** Aufgaben wie Überarbeitungen, Entfernungen, Ersetzen, Aktivieren oder Deaktivieren von Funktionen und jede andere technische Aufgabe