

# **Reporting Bugs**

Kommunikation mit Entwicklern freier Software  
Jochen Hein

Copyright © 2003 Jochen Hein

# **Reporting Bugs**

**Was sind eigentlich Bugs?**

**Ich habe einen Bug gefunden. Was nun?**

**Kontakt zum Projekt**

**Bug Tracking**

**Lohnen sich Bug-Reports?**

**Kommerzielle Software hat keine Bugs**

**Bug Hunting im großen Stil**

**Fazit**

## **Was sind eigentlich Bugs?**

**Ein Programm bricht mit oder ohne Meldung ab, möglicherweise mit einem Core-Dump.**

**Das Programm tut nicht das, was in der Dokumentation steht.**

**Die Dokumentation ist falsch, unvollständig oder hilft nicht weiter.**

**Eine Übersetzung ist unvollständig, unverständlich oder falsch.**

**Das Programm tut nicht das, was ich als Benutzer erwarte oder ist nur schwer zu bedienen.**

**Das Programm hält sich nicht an Style-Guides oder Standards.**

## **Ich habe einen Bug gefunden. Was nun?**

**Betrifft der Bug die neueste Version? Findet sich im ChangeLog ein interessanter Eintrag?**

**Kann ich einen fertigen Verbesserungsvorschlag erstellen? Gerade bei Dokumentation muss man nicht programmieren können!**

**Informationen sammeln und aufbereiten: Release, Systemumgebung, Patches, Protokolle, Erwartungen und Ergebnisse. Eventuell Interpretationen.**

**Nachlesen, wie Bugs gemeldet werden können oder sollen. Das ist je Projekt unterschiedlich.**

## **Kontakt zum Projekt (Tools)**

**Direkten Kontakt zum Entwickler aufnehmen (meist per Mail).**

**Bugtracking mittels Bugzilla oder ähnlichen Tools.**

**Spezielle Mail-Adressen oder Newsgroups.**

**Programme wie gnats, send-pr oder reportbug.**

**Online-Diskussionen im IRC (Internet Relay Chat).**

## **Kontakt zum Projekt (Inhalt)**

**Stöbern in Archiven: Wie werden Bug-Reports bearbeitet?**

**Welche Art von Reports kommen besonders gut an?**

**Bugzilla: Recherche in alten Bugs. Vermeidung von doppelten Reports.**

**Höflich und freundlich - die meisten Entwickler machen das nur aus Spaß.**

**Wenn das Bug-Tracking nicht öffentlich ist: nachfassen.**

**Asking Smart Questions: <http://www.tuxedo.org/~esr/faqs/smart-questions.html>**

**GNU-Programme haben manchmal eine Anleitung für Bug-Reports.**

## **Bug-Reporting ist kompliziert. Lohnt sich das?**

**Ja, es lohnt sich!**

**Keine Marketingabteilung, direkter Kontakt zum Entwickler**

**viele Beta-Tester: "Given enough eyeballs, all bugs are shallow"**

**Ein behobener Bug steht hoffentlich einem Neu-Einsteiger nicht im Weg.**

**Man lernt Entwickler und andere Anwender kennen.**

**Mit Bug-Reports kann auch ein Nicht-Programmierer etwas für die Community tun.**

**Bugfixes führen zu mehr Qualität, eventuell durch automatische Tests und Peer-Review der Korrekturen.**

## **Mein Bug wird nicht gefixt. Warum?**

**Zu wenige oder falsche Informationen**

**Zu wenig Erklärung zum eigenen Umfeld**

**Der Entwickler hat keine Zeit und ertrinkt in Bugs.**

**Es ist kein Bug.**

**Code talks. Wenn man eine genaue Programmänderung vorschlagen kann und diese begründen kann, dann steigen die Chancen.**

**Testfälle helfen. Der Fehler (und die Funktion der Korrektur) werden offensichtlich.**



## **Kommerzielle Software hat keine Bugs**

**Bugs werden durch eine Hotline bearbeitet. Diese beseitigt oft das Symptom, nicht aber die Ursache.**

**Feedback und Erweiterungswünsche werden durch das Marketing oder die Hotline gefiltert. Entwickler sitzen im "stillen Kämmerlein" ohne Kontakt nach außen.**

**Bearbeitung von Bugs kann sehr lange dauern (z.B. sicherheitskritische SAP R/3 Bugs über sechs Jahre - immer noch offen).**

**Anwender können sich in der Regel nicht selbst und auch nicht untereinander helfen.**

**Anwender haben keine Möglichkeit alleine die Ursache des Problems zu erforschen und Fixes zu verteilen.**

# **Bug-Hunting im großen Stil**

**Bug Squashing Party**

**GNOME Bug Day on IRC**

**Hack-fest**

## **Fazit**

**Bugs sind lästige kleine Biester, die immer wieder auftauchen.**

**Bug-Hunting kann Spaß machen, ist aber anstrengend.**

**Gefixte Bugs sind ein echter Erfolg.**

**Auch Kosmetik (Dokumentation, Tippfehler) ist nützlich.**

**Die Software wird besser in Bezug auf Security, Stabilität, Performance, Flexibilität.**

**The only problem with trouble shooting is that the trouble shoots back.**