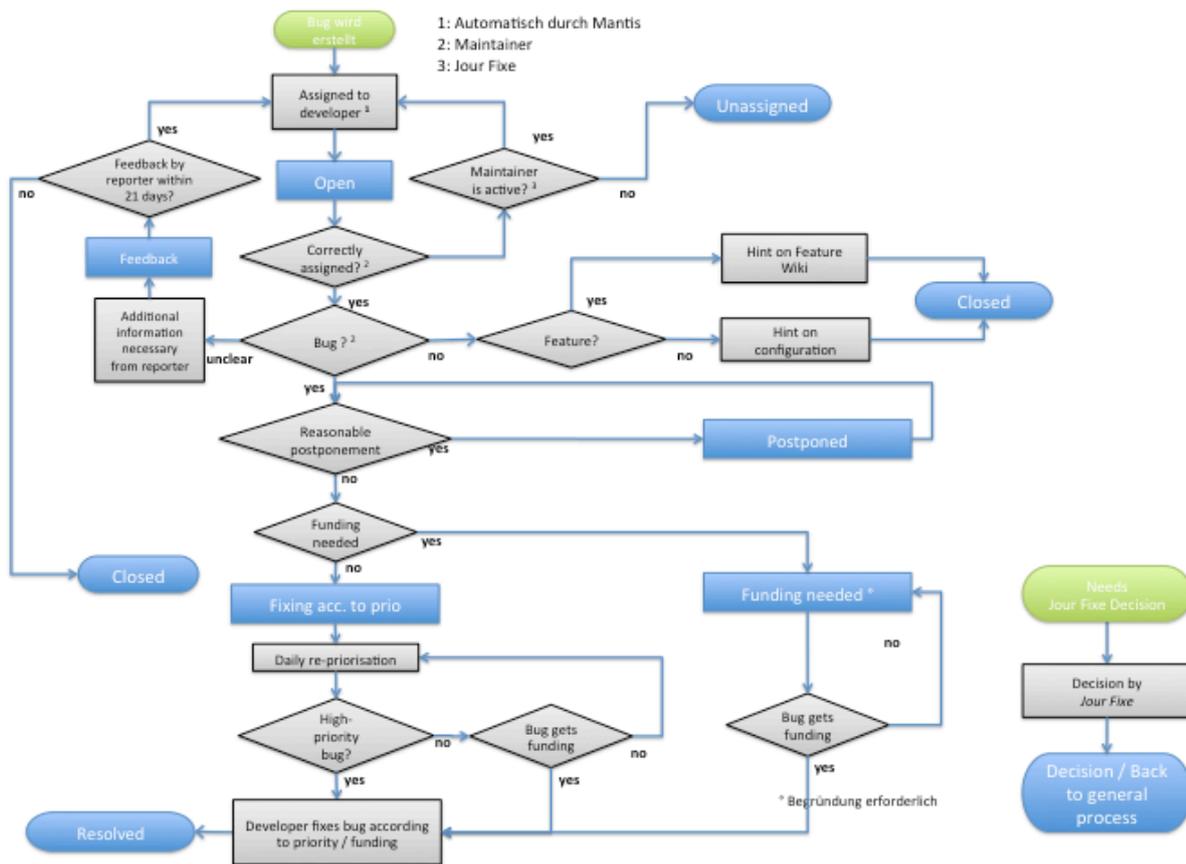


# ILIAS – Bugfixing-Prozess



## 1 Hintergrund

Auf der letzten Vereinssitzung am 28. März 2014 wurde folgender Antrag angenommen.

Der Vorstand wird beauftragt grundsätzliche Regeln zu definieren, wie Fehler die über den Mantis-Bugtracker gemeldet werden/wurden, behandelt werden. Ziel sollte dabei eine zeitnahe Behebung der Fehler sein. Insbesondere muss eine Regel für den Umgang und Bekanntmachung von komplexen Fehlern, die nicht zeitnah behoben werden können, getroffen werden. Diese Regeln müssen klar die Pflichten eines Maintainers umschreiben.

Dieses Dokument stellt das von der Mitgliederversammlung gewünschte Regelwerk dar. Es ist das Ergebnis eines Prozesses von mehreren Vorstandssitzungen, intensiven Diskussionsrunden im ILIAS-Forum, sowie mehreren Präsenztreffen mit den Service Providern bzw. Maintainern.<sup>1</sup>

## 2 Ausgangssituation

Wie in jeder Software existieren auch in ILIAS Programmfehler, sogenannte Bugs. Eine vollständig bug-freie ILIAS-Software zu schaffen, ist nicht möglich. Angestrebt werden kann nur, die Zahl der Bugs so niedrig zu halten, dass eine Nutzung der Software durch die Anwender gut möglich ist und keine schwerwiegenden Fehler auftreten.

Eine zentrale Aufgabe des ILIAS-Vereins ist es, die Rahmenbedingungen für eine kontinuierliche Qualitätssicherung zu schaffen. Der Verein kann aber aktuell nicht als Dienstleister fungieren, der in Eigenregie Fehler im Programmcode behebt.

Vom Verein können Anwender von ILIAS folgende Leistungen erwarten. Der Verein

- kümmert sich um den Rahmen des Bugfixing-Prozesses
- vermittelt die Interessen aller am Prozess Beteiligten
- verleiht über Entwicklungsleiter und Produktmanager die Modul-Maintainerschaften an Entwickler
- koordiniert mithilfe von Entwicklungsleiter und Produktmanager den Bugfixing-Prozess
- stellt die für den Prozess notwendigen Tools zur Verfügung (zurzeit: Bugtracker Mantis, Test Case Management-Tool Testrail und Test-Installationen pro Major-Release)

**Wichtig:** Der Verein hat keine Weisungsbefugnis auf die Entwickler (z.B. hinsichtlich der Einhaltung von Fristen beim Bugfixing). Dieses Dokument basiert jedoch auf einem Konsens, den der ILIAS-Vorstand mit den Entwicklern erarbeitet hat.

<sup>1</sup> Der Vorstand dankt an dieser Stelle den zahlreichen aktiven Vereinsmitglieder.

### 3 Modul-Maintainerschaft

Gemäß ILIAS-Entwicklungsprozess gibt es für jede Software-Komponente einen namentlich benannten Modul-Verantwortlichen – auch Maintainer genannt. Dieser kümmert sich neben der Weiterentwicklung auch um die Behebung von Fehlern in der jeweiligen Komponente.

Die Regeln des ILIAS Development Guides dienen hierbei als Grundlage für Maintainer. Deren Rechte und Pflichten werden unter [www.ilias.de/docu/goto\\_docu\\_pg\\_33026\\_42.html](http://www.ilias.de/docu/goto_docu_pg_33026_42.html) ausführlich beschrieben. Zentrale Aspekte sind dabei:

- Der Modul-Maintainer übernimmt die Pflege des von ihm betreuten Software-Moduls für mindestens drei Jahre.
- Der Modul-Maintainer koordiniert die Weiterentwicklung seiner Komponente in Absprache mit Entwicklungsleiter und Produktmanager.
- Der Modul-Maintainer ist verantwortlich für das Bugfixing seiner Komponente und erhält automatisiert die gemeldeten Fehler seines Moduls zugewiesen.

Das Bugfixing der eigenen Komponenten übernimmt ein Modul-Maintainer in der Regel kostenlos. Im Gegenzug dazu ist er erster Ansprechpartner für weitere Entwicklungen seines Moduls. Der Modul-Maintainer erklärt sich zudem bereit, Fehler im Rahmen seiner Möglichkeiten zeitnah zu beheben und den definierten Bugfixing-Prozess einzuhalten.

#### Herausforderungen für Modul-Maintainer:

Bei der Regelung des Bugfixings muss darauf geachtet werden, dass die Übernahme einer Modul-Maintainerschaft nicht mit großen Hürden verbunden ist und damit für Entwickler und Serviceprovider unattraktiv wird. Zu berücksichtigende Probleme für Maintainer sind hierbei:

- **Module mit geringem Veränderungsbedarf:** Einige Module und Funktionen werden nur im geringem Maße weiterentwickelt. Damit ist eine Gegenfinanzierung des Bugfixings durch Umsätze für Weiterentwicklungen des jeweiligen Moduls nicht möglich.
- **Übernahme von Altlasten:** Tritt ein Modul-Maintainer von seiner Aufgabe zurück oder verlässt das Entwicklerteam, muss zeitnah ein neuer Maintainer gefunden werden. Dieser steht vor der großen Herausforderung, sich rasch in das neue Modul einzuarbeiten und den Code des bisherigen Maintainers mitzubetreuen. Ein kostenloses Bugfixing von Software-Code seines Vorgängers ist damit eine risikobehaftete Investition des neuen Modul-Maintainers bzw. Serviceproviders.

## 4 Ziele / Qualitätsmerkmale

Alle an der Diskussion beteiligten sind sich darin einig, dass die obersten Ziele des Bugfixing-Prozesses die ständige und nachhaltige Verbesserung der Produktqualität von ILIAS und das Erreichen einer höheren Kundenzufriedenheit sind.

### 4.1 Ziele des Bugfixing-Prozesses

Um diese Ziele erreichen zu können, muss ein Bugfixing-Prozesses folgende Kriterien erfüllen:

1. Der Prozess muss **verständlich** und **transparent** sein.
2. Die Bug-Behebung muss **planbar** sein.  
Der Bug wird gemäß der im Bug-Status definierten zeitlichen Fristen behoben.
3. Es existiert ein **klar geregelter Kommunikationsfluss**:  
Gemäß der im Prozess definierten zeitlichen Fristen wird zwischen den Beteiligten kommuniziert.
4. Die Durchführung des Bugfixing-Prozesses erfolgt mit möglichst **geringem Overhead** hinsichtlich der Zeit und Kosten
  - Keine Überlastung des Jour Fixes
  - Automatische Abläufe / Benachrichtigungen wo möglich
  - Festgelegte Kriterien für Bug-Reports
5. Der Prozess ist **ressourceneffizient**  
Die Kapazität der Programmierer soll durch Priorisierung auf gravierende Bugs gelenkt werden.
6. Der Prozess schafft **keine zusätzlichen Hürden** bzw. Barrieren für die Übernahme einer Modul-Maintainerschaft.
7. **Bug-Fixing und Test-Case-Erstellung** müssen eng verzahnt sein.

### 4.2 Qualitätsmerkmale für den reibungslosen Ablauf des Bugfixing-Prozesses

Neben den o.a. Zielen zeichnet sich aus Sicht des Vereins ein qualitativ zufriedenstellender Bugfixing-Prozess durch folgende Merkmale aus.

#### 4.2.1 Zeitliche Reaktionen

Die Reaktionszeiten sind bei bestimmten Prozessschritten definiert, so dass alle Beteiligten

wissen, in welchen Fristen eine Reaktion zu geben und zu erwarten ist.

#### 4.2.2 Klare Statuszuweisung je Bug

Für jeden Bug existiert eine klare Statuszuweisung, die von allen Beteiligten auch einheitlich verstanden und angewandt wird.

Status	Beschreibung	Zuweisung und Zeitpunkt / Fristen
<b>Open</b>	Bug ist dem zuständigen Maintainer zugewiesen. Dieser wird den Report binnen 21 Tagen <sup>2</sup> anschauen und mit neuem Status versehen.	Automatisierte Zuweisung direkt bei Anlage des Reports durch Mantis
<b>Unassignend</b>	Bug kann keinem aktiven Maintainer zugewiesen werden. Beispiele hierfür sind Fehler, die auf sehr altem Code basieren oder Code, für die kein Maintainer zuständig ist.  Hier kann aktuell kein Bugfix angeboten werden.	Zuweisung durch Jour Fixe (14-tägig) bzw. Entwicklungsleiter
<b>Feedback</b>	Maintainer benötigt vom Reporter weitere Informationen, um Bug zu beurteilen und zu fixen.	Zuweisung durch Maintainer spätestens 21 Tage nach Statuszuweisung „Open“  Laufzeit für Feedback: 21 Tage
<b>Fixing According to Priorisation</b>	Bug wird vom zuständigen Maintainer in Abhängigkeit von der Priorisierung durch die Vereinsmitglieder gefixt.	Zuweisung durch Maintainer spätestens 21 Tage nach Statuszuweisung „Open“
<b>Postponed</b>	Behebung des Fehlers wird vom Maintainer mit einer ausreichenden Begründung zurückgestellt, da entweder eine Überarbeitung der Komponente im nächsten Major-Release ansteht oder ein anderer triftiger Punkt besteht.	Zuweisung durch Maintainer spätestens 21 Tage nach Statuszuweisung „Open“
<b>Funding Needed</b>	Bug wird vom zuständigen Maintainer anerkannt, kann aber nicht im Rahmen eines kostenlosen Bugfixes zeitnah behoben werden (Hinweis auf Funding in Bugreport).	Zuweisung durch Maintainer spätestens 21 Tage nach Statuszuweisung „Open“
<b>Needs Jour Fixe Decision (NJFD)</b>	Vor einer weiteren Bearbeitung des Bugs ist eine Entscheidung über den Charakter des Bugs bzw. über eine mögliche Lösung notwendig.	Zuweisung durch Reporter oder Maintainer  Behandlung des Bugreport binnen 14 Tagen auf nächsten Jour Fixe
<b>Resolved</b>	Bug wurde behoben und tritt nicht mehr auf. Im Bugreport wird angegeben, in welchem Release der Bugfix bereitsteht (und ggfs. weitere Infos zur Art des Bugfixes)	Zuweisung durch Maintainer nach Commit des Bugfixes in Mantis

<sup>2</sup> 21 Tage werden nachfolgend immer als Kalendertage verstanden, entsprechend 3 Wochen

<b>Closed</b>	Bugreport wurde geschlossen Im Bugreport wird der Grund angegeben	Zuweisung durch Maintainer, (a) wenn es sich um keinen Bug sondern um ein Feature handelt, (b) wenn Reporter nicht binnen 21 Tagen ein Feedback gibt.  Zuweisung durch Reporter
---------------	----------------------------------------------------------------------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

#### 4.2.3 Einheitliche Beschreibung der Schwere eines Fehlers

Die nachfolgenden Angaben basieren auf der bereits praktizierten Skala aus dem Development Guide-Kapitel: „Getting Involved as Tester“ ([www.ilias.de/docu/goto\\_docu\\_pg\\_9809\\_42.html](http://www.ilias.de/docu/goto_docu_pg_9809_42.html)). Um die Zuweisung durch die Reporter zu vereinfachen, wurden sowohl die beiden schwersten Ausprägungen (block und crash) als auch die beiden leichtesten (tweak und typo) jeweils zu einer Stufe zusammengefasst.

Ausprägung	Beschreibung	Beispiel
<b>Block</b>	Es ist nicht möglich, ILIAS zu nutzen. Oder es liegt eine gravierende Verletzung der IT-Sicherheit oder des Datenschutzes vor. Es gibt keinen Workaround.	Server stürzt ab oder generelle Nutzung von ILIAS nicht möglich.
<b>Major</b>	Die betroffene Komponente ist nicht oder nur stark eingeschränkt nutzbar. Dagegen können andere Teile des Systems weiter genutzt werden.	Im Forum können keine Beiträge mehr angelegt werden.
<b>Minor</b>	Die betroffene Komponente ist fehlerhaft, kann aber mit kleineren Einschränkungen genutzt werden. Auch andere Teile von ILIAS sind weiterhin	Die Sortierung der Forenbeiträge ist fehlerhaft.
<b>Tweak</b>	Die Nutzung von ILIAS ist nicht eingeschränkt. Allerdings stört der Fehler im alltäglichen Gebrauch.	Falsch positionierter Button, inkonsistentes Layout oder Tippfehler in einer Sprachvariablen

#### 4.2.4 Kriterien einer qualitativ guten Bugmeldung

Eine möglichst genaue Beschreibung des Fehlers unterstützt den Modul-Maintainer bei der zeitnahen Behebung von Bugs. Je besser ein Bug berichtet wird, desto einfach kann dieser nachgestellt und ggf. behoben werden.

Als Mindestangaben für die Meldung eines Bugs gelten:

Information	Erläuterung	Form / Ausprägungen
<b>Summary</b>	Vergabe eines Titels für den Bug	Eingabe Freitext
<b>Category</b>	Angabe der Komponente, in der der Fehler auftritt	Vordefinierte Auswahlliste der ILIAS-Komponenten, z. B. „Blog“

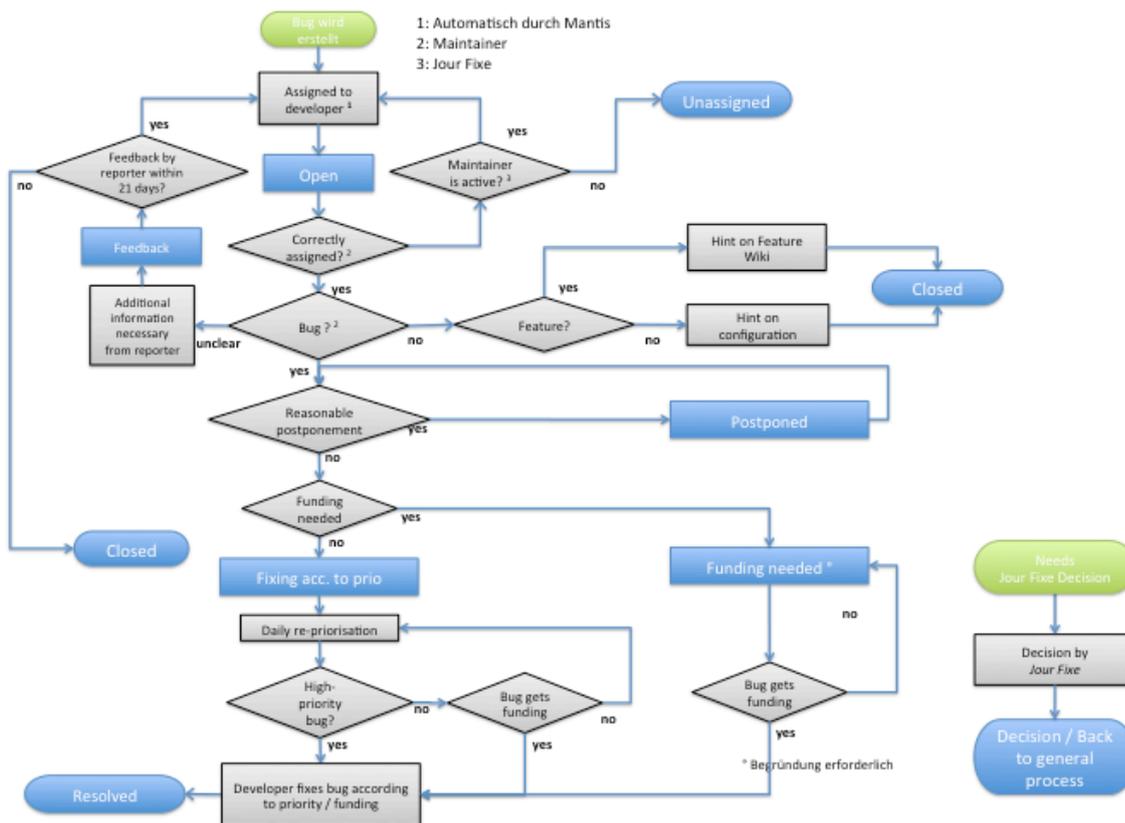
<b>Severity</b>	Einstufung des Schweregrads des Fehlers	Vordefinierte Auswahlliste: <ul style="list-style-type: none"> <li>• tweak</li> <li>• minor</li> <li>• major</li> <li>• block</li> </ul>
<b>Version</b>	Angabe der betroffenen Version	Vordefinierte Auswahlliste der Releases, z. B. „4.4.1“
<b>Reproduced on</b>	Angabe zur Installation, auf der Fehler auftritt	Vordefinierte Auswahlliste: <ul style="list-style-type: none"> <li>• ILIAS test installation</li> <li>• own installation only</li> <li>• n/a</li> </ul>
<b>Reproducibility</b>	Angabe zur Reproduktion des Fehlers	Vordefinierte Auswahlliste: <ul style="list-style-type: none"> <li>• have not tried</li> <li>• unable to reproduce</li> <li>• random</li> <li>• sometimes</li> <li>• always</li> </ul>
<b>Description</b>	Beschreibung des Fehlers und wie er auftritt, sowie Fehlermeldung (falls ausgegeben) Bei Bugreports im Rahmen der Softwaretests auf Basis Test Cases aus Testrail sowie der dort dokumentierten Eigenschaften	Eingabe Freitext

Darüber hinaus kann der Bugreport noch durch folgende Informationen verbessert werden:

- Screenshot, der fehlerhaftes Verhalten aufzeigt
- Nachstellen des Fehlers auf ILIAS-Test-System und Angabe des PermaLinks
- Abfilmen der Vorgehensweise
- Falls Fehler nicht auf Test-Installation reproduzierbar ist, weitere Angabe von Einstellungen, die auf betroffenen System zum Zeitpunkt des Fehlers aktiviert waren (z. B. „Öffentlicher Bereich aktiviert“ oder „Oracle als Datenbank“)

## 5 Bugfixing-Prozess

Nachfolgend werden die verschiedenen Teilprozesse des überarbeiteten Bugfixing-Prozesses von ILIAS beschrieben.



### 5.1 Bug berichten

Ein Benutzer (Reporter) meldet einen Bug, der in ILIAS auftritt, im zentralen Bugtracker [www.ilias.de/mantis](http://www.ilias.de/mantis). Der Bugreport wird entsprechend der Vorgaben (siehe Kap. 4.2) erstellt.

Nach Speichern des Bugreports weist Mantis den Bug automatisch dem Entwickler zu, der für die betroffene Komponente zuständig ist (= Maintainer). Der Bugstatus wird automatisch auf **Open** gesetzt. Der Maintainer erhält eine E-Mail-Benachrichtigung und hat **21 Tage** Zeit, zu reagieren<sup>3</sup>. Erfolgt keinerlei Reaktion, kann der Reporter den Jour-Fixe anrufen.

<sup>3</sup> Reagieren bedeutet im vorliegenden Fall, den nächsten Schritt im Bugfixing-Prozesses zu ergreifen.

## 5.2 Bugzuordnung überprüfen

Ein Entwickler, dem ein Bug zugewiesen wurde, überprüft bei der ersten Sichtung des Bugreports, ob dieser Bug in seiner Verantwortung liegt oder ob ein anderer Entwickler für diesen zuständig ist. Wird der Bug einem anderen Maintainer zugewiesen, erhält dieser eine Benachrichtigung. Der Bugstatus bleibt auf **Open** und die 21-Tage-Frist wird neu gesetzt.

Ist kein Maintainer für diese Komponente zuständig (z. B. weil der Fehler auf sehr altem Code basiert) oder der Maintainer nicht mehr aktiv, wird der Bugstatus vom Entwickler auf **Unassigned** gesetzt.

Bei Unklarheit kann der Entwickler den Jour Fixe um Entscheidung bitten. Der Jour Fixe wiederum kann in seinen **14-tägigen** Sitzungen die korrekte Zuweisung der Bugs überprüfen.

## 5.3 Feedback

### 5.3.1 Feedback vom Reporter einfordern

Fehlen dem Entwickler Informationen für die Beurteilung bzw. Bearbeitung des Bugreports, kann dieser weitere Angaben vom Reporter einfordern. Hierzu ändert er den Bugstatus auf **Feedback** und stellt dem Reporter entsprechende Fragen.

### 5.3.2 Feedback durch Reporter geben

Wird der Reporter vom Entwickler um Feedback gebeten, erhält der Reporter automatisch eine Benachrichtigung von Mantis. Der Reporter hat nun **21 Tage** Zeit, die gewünschten zusätzlichen Informationen in den Bugreport einzutragen und den Bug wieder dem Entwickler zuzuweisen. Der Bugstatus wird dabei wieder auf **Open** gesetzt.

### 5.3.3 Bug schließen wegen fehlendem Feedback

Gibt der Reporter kein Feedback, kann der Entwickler den Bugreport nach **21 Tagen** schließen. Dazu setzt er den Bugstatus auf **Closed** und gibt als „Resolution“ **Suspended** an.

## 5.4 Bug oder Feature?

Ein Entwickler, dem ein Bug zugewiesen wurde, überprüft im nächsten Schritt, ob der Report tatsächlich einen Bug adressiert oder ob es sich um ein Feature handelt.

- Fordert der Report eine funktionale Erweiterung einer bestehenden Funktion, kann der Entwickler den Report als Feature-Request einstufen und mit einem Verweis auf das Feature Wiki schließen.

- Behandelt der Report eine offensichtliche Fehlkonfiguration von ILIAS, kann der Entwickler den Report mit einem Hinweis auf die korrekte Konfiguration oder die Installationsanleitung schließen.

In beiden Fällen wird der Bugstatus auf **Closed** gesetzt.

## 5.5 Begründete Verschiebung

Liegen dem Entwickler ausreichende Informationen vor, kann dieser den Bug dahingehend bewerten, ob er in nächster Zeit und für die aktuell gewarteten Releases gefixt werden kann oder ob es einen triftigen Grund gibt, das Bugfixing zu verschieben.

Liegt ein triftiger Grund für eine Verschiebung vor, setzt der Maintainer den Bug auf **Postponed**.

Triftige Gründe können sein:

- Komponente wird zur nächsten Version komplett überarbeitet und Bug wird in diesem Rahmen bearbeitet.
- Die weitere Entwicklung der Komponenten wird nicht mehr unterstützt.
- Der Bug tritt in einer ILIAS-Version auf, die nicht mehr supportet wird.
- Weitere Gründe

## 5.6 Finanzierung

Unabhängig von der vorliegenden Regelung kann ein Auftraggeber eines betroffenen Features aufgrund seiner vertraglichen Ansprüche davon ausgehen, dass ein Bug innerhalb der Gewährleistungsfrist vom Auftragnehmer kostenlos behoben wird. Für Bugs, die nach Ende der Gewährleistungsfrist gemeldet werden, besteht kein Anspruch auf kostenloses Bugfixing (vgl. GPL 3).

### 5.6.1 Finanzierung für Bugfix notwendig

Sind für das Bugfixing umfangreiche Arbeiten am Quellcode notwendig, die über das übliche Maß<sup>4</sup> der Fehlerbehebung hinausgehen, kann der Entwickler den Bug auf **Funding Needed** setzen und um eine Finanzierung des Bugfixings bitten.

### 5.6.2 Finanzierung für Bugfix gesichert

Erhält der Entwickler eine Zusage für die Finanzierung des Bugfixes, kann dieser den Bug zeitnah beheben und den Bugfix unter Angabe der Mantis-Nummer committen.

<sup>4</sup> Als Arbeiten, die das übliche Maß überschreiten, werden verstanden: Aufwändige Reproduktion oder Analyse des gemeldeten Problems; Mit-/Hauptursächlichkeit der Vorentwicklungen eines anderen Entwicklers; konzeptionelle Fragestellungen.

Anschließend setzt der Entwickler den Bugreport auf den Status **Resolved** und nennt die Version, für die der Bug gefixt wurde.

## 5.7 Priorisierung der zu behehenden Bugfixes

Ist keine Finanzierung notwendig, um den Bug zu beheben, setzt der Entwickler den Bug auf Status **Fixing According to Priorisation**. Damit wird der Bug automatisch in die Liste der priorisierten Bug-Reports gesetzt.

Jedes Vereinsmitglied kann jederzeit die Priorität eines bestehenden Bugs mit Status **Fixing According to Priorisation** hoch- oder wieder herabsetzen. Als Optionen werden die Punktwerte **0** und **1** angeboten. Jedes Vereinsmitglied erhält damit die Möglichkeit, das Interesse an der Bug-Behebung mit einem Punkt zu signalisieren. Eine Übersicht in Mantis zeigt die Bugs in der Reihenfolge der vergebenen Punkte an. Je mehr Punkte ein Bug hat, desto schneller sollte er behoben werden.

Gehört ein vorliegender Bugreport zu den hoch-priorisierten Bugs des Entwicklers, behebt dieser diesen zeitnah und committed den Bugfix unter Angabe der Mantis-Nummer.

Anschließend setzt der Entwickler den Bugreport auf den Status **Resolved** und nennt die Version, für die der Bug gefixt wurde.

## 5.8 Paralleler Weg: Forcierung des Bugfix durch Finanzierungszusage

Wie oben beschrieben, gilt das Commitment, dass ein Modul-Maintainer vorhandene Bugs gemäß des in diesem Dokument beschriebenen Prozesses behebt. Dabei liegt es im seinem Ermessen, wie viel Zeit er für das Bugfixing investiert.

Sollte die Priorisierung durch die Community nicht den Wünschen des Bug-Reports entsprechen, so kann jederzeit ein Bug auf der Grundlage eines Vertrags mit dem Modul-Maintainer (gegen Entgelt) forciert werden.

Bietet ein Anwender für einen nicht ausreichend hoch priorisierten Bug eine Finanzierung an, kann der Entwickler den Bug zeitnah beheben und den Bugfix unter Angabe der Mantis-Nummer committen. Anschließend setzt der Entwickler den Bugreport auf den Status **Resolved** und nennt die Version, für die der Bug gefixt wurde.

Besteht keine Finanzierungszusage, verbleibt der Bug in der Liste der zu priorisierenden Bugfixes.

## 5.9 Anrufung Jour Fixe

Entwickler und Reporter können bei Unklarheiten oder Unstimmigkeiten jederzeit den Jour Fixe um eine Entscheidung bitten. Dazu wird das Anliegen bzw. die Bitte um eine Entscheidung kurz

im Bugreport beschrieben, der Status in **Needs Jour Fixe Decision** geändert und der Bugreport auf die Agenda des kommenden Jour Fixes gesetzt.

Der Jour Fixe fällt dann eine Entscheidung, veröffentlicht diese im Bugreport und im Protokoll der Sitzung innerhalb des Feature-Wikis und ändert entsprechend den Status des Bugs.

\* \* \*