

Was ist eine Beta-Version ?

Aus Erfahrung als langjährig in der US-Software-Entwicklung bei großen Firmen tätig, vielleicht 'ne [Erklärung](#) -nicht nur zu Beta-Versionen.

Um erst mal den **Umfang** eines mittelprächtigen Programmes zu verdeutlichen:
Ein Star-Programmierer schafft es, jeden Tag etwa **30 Blätter DIN-A4** beidseitig eng vollzuschreiben mit äußerst sinnvollem Text und mit Referenzen auf andere Seiten, die vor Monaten schon von anderen geschrieben wurden - **fehlerfrei**.
Da kommen schon mal leicht so 100-150 Tausend Zeilen zusammen, fehlerfrei.

Und zwar über ein-zwei Jahre täglich, ohne Urlaub, ohne Feiertage, durchgehend.
Und es sitzen eine Handvoll Leute an so einem Programm.
Nicht alle brilliant, manche sogar grottenübel.

Man kann so einen Prozess am besten vergleichen mit dem Schreiben der Bibel.
Ein einziger Fehler kann alles zerstören.

Vorab:

Das hier beschriebene Verfahren ist die total FALSCH E ENTWICKLUNG,
aber leider ist das die Wirklichkeit.

[Essenz:](#) (siehe auch Erklärung am Ende)

Erst mal Finger weg von einer Beta-Version.

Es gibt drei (vier ... ?) Stufen, Abschnitte in der Software-Entwicklung

1) **Inhouse** (intern), das heißt, daß die Programme während der laufenden Entwicklung schon getestet werden. Das geschieht bröckchenweise, funktionsweise. Einzelne Funktionen werden intensiv getestet.
So ein Programm kommt niemals in den Handel. Sollte es jedenfalls nicht.

2) Es entsteht so langsam eine **Alpha**-Version, auf die man 'stolz' sein kann. Diese Version geht an große, befreundete Firmen, Unis, ...
Und die Jungs dort testen ohne Gnade und ohne irgend ein belastendes Vorwissen das gesamte Produkt auf Tüchtigkeit. Danach vergeht der Stolz schon mal.

3) Parallel dazu kommen **Wünsche** der beteiligten Partner ans Licht, die jetzt noch nachprogrammiert werden. Z.B. wird ein bestimmtes Layout gewünscht.
Kann schon schwierig sein.

4) Wenn das nach vielem (monate-)langem Hin-und-Her gut ausgeht, dann wird eine **Beta**-Version rausgelassen. An befreundete Unternehmen.
Und diese **Beta**-Version geht auch schon mal in den Handel - mit dem deutlichen Hinweis, daß dort (immer noch) Fehler drin sein können.

Nach einer gewissen Periode wird dann angenommen, daß alles im Reinen ist, d.h. kein (wesentlicher!) Fehler mehr zu erwarten ist.
Das ist die **Verkaufsversion**.

Was aber KEINESFALLS automatisch bedeutet, daß dort kein Fehler mehr drin ist. Die Verkaufsversion kommt auch schon mal 'zwangsweise' in den Handel, wenn nämlich die Verkaufsabteilung, oder die Shareholder (Aktionäre), das Budget, eine **deadline** (einen Stichtag) setzten.
Sogar egal wenn noch Fehler bekannt sind.

Ergebnis:

Wir erleben es immer wieder > updates kommen regelmäßig.
Meistens Sicherheitsupdates. Traurig bis bedenklich die Sache.

Warum ist das so?

Ein Softwareprodukt ist auch eine finanzielle Angelegenheit. Das bedeutet, daß zur Entwicklung ein bestimmter Betrag vorgesehen ist, investiert wird. Der reicht natürlich nicht endlos lange. Man setze den Betrag dreifach an.

So entsteht -besonders gegen Ende der Entwicklung- der Druck, das Produkt endlich auf den Markt bringen zu müssen.

Und 95% der Entwicklung sind einfach und schnell; die 'restlichen' 5% kosten einen immensen weiteren Aufwand. Besonders wenn die Grundlage schon starr verkorkst angelegt wurde. Ähnlich der Korrektur eines bereits gedruckten Buches.

Es ist nun aber so, daß die Entwicklung umfangreicher Programme SEHR sorgfältig geplant und ausgetüftelt werden muß - bevor es richtig losgeht.

Und dazu gehört nicht nur ein tüchtiger Softwaremensch, sondern besonders jemand, **der den totalen Überblick hat und behält.**

Es ist nicht gerade einfach, 20 und mehr Entwickler unter einen Hut zu kriegen. Mit all ihren persönlichen 'Macken'. Mancher kündigt, mancher kommt hinzu mit gänzlich anderem Programmierstil; so wie es auch unterschiedliche Romanstile gibt. Und wenn eine **Schlüsselfigur** den Abgang macht, stirbt ein umfangreiches Produkt schon mal sang- und klanglos. Andere werden dann bis zum Ende durchgewürgt - und sehen dann auch so aus.

Abgesehen noch davon, daß auch oft genug einfach technische Mißverständnisse vorkommen, wenn nämlich jemand etwas anderes verstanden hat, als es der andere sagte - oder Vorgaben falsch ausgelegt hat. Auch ein täglich im ST auftretendes Phänomen.

Generell ist der Ansatz, Programme 'einfach so', nach sturen Vorgaben zu schreiben, schon mal falsch.

Nämlich sich bei der Entwicklung schon darauf zu verlassen, daß der Tester schon die (Tipp-) Fehler finden wird, ist absolut tödlich.

Ähnlich einem Autor, der sich darauf verlässt, daß der Lektor schon seine Fehler finden -und korrigieren und Passagen hinzufügen- wird.

Abgesehen davon daß es auch lausige Tester gibt.

Jedoch wird Software weltweit so geschrieben - leider.

Der Grund dafür ist immer wieder der Kostendruck - und damit die Zeit.

Beta-Versionen zeugen also davon, daß nicht mal der Hersteller daran glaubt, das Ding fehlerfrei hingekriegt zu haben. Schwach.

Und nun durch die Kennzeichnung 'Beta' nicht allzuviel Dresche zu kriegen. Siehe dazu auch unten den Autobauer.

Fehlerfreie Software zu schreiben IST jedoch möglich

(auch wenn immer das Gegenteil behauptet wird -das sind faule Ausreden-).

Aber es dauert extrem lange; und es braucht extrem gute, erfahrene Programmierer - die sehr sehr selten (und teuer, aber letztlich preiswert) sind.

Fehlerfreie und gute Programme sind deshalb genau so selten wie exzellente Programmierer.

Anekdote:

Aus meiner Laufbahn kenne ich eine einzige Frau !, die das geschafft hätte, wäre die Firma nicht vorher eingegangen.

Und diese Frau wurde angefeindet aus allen Ecken.

Niemand in dieser 200-Mann-Klitsche kam mit ihr zurecht; auch ich zuerst nicht.

Also ein zusätzliches persönliches Problem zur technischen Seite.

Im Endeffekt hatte sie jedoch recht; denn die Firma ging ein, weil ihr eben nicht gefolgt wurde. Und mit dieser Firma ging die noch größere 'Mutter' auch ein. Namen kann ich nicht nennen, es war die zweitgrößte Computerfirma der Welt.

Seiteneffekt:

Der Kostendruck -und die Absicht, endlich mit dem Programm ordentlich Kasse zu machen- hat auch einen interessanten Seiteneffekt.

Nämlich daß 'hingerotzte' Programme endlos lange Kopfschmerzen bereiten mit immer wieder neuen unerwarteten Fehlern in der Praxis bei Kunden.

Letztendlich geht daran dann diese Firma ein, weil niemand das Produkt kaufen wird (und unter Umständen auch noch die Firma, die das Programm nutzen wollte).

Jeder Verkauf bürdet dieser Firma auch noch so viele Reklamationen auf, daß sie gar nicht mehr dazu kommt, die Fehler alle zu beheben. Geschweige denn an Weiterentwicklungen zu denken.

Von Anfang an in den Sand gesetzt dank falsch angesetzter Programmiermethoden. **Nämlich sich darauf verlassend, dass der Tester (inhouse, Alpha, Beta, ...) das schon richten wird.**

Tödlich falsch!

Aber genau so entstehen Beta-Versionen.

Auch bekannt als '**Bananen-Software**' > reift beim Kunden.

Man beachte den Zufall **B**eta und **B**anane.

Kommt mir so vor wie jener Autobauer:

Hallo, wir haben ein ganz schnittiges neues Modell entwickelt; wirklich wunderbar - muss jeder haben.

Nur wissen wir nicht ob die **Bremsen auch schon funktionieren.**

Berichtet also mal.

Facit, Empfehlung:

Nur wer ausreichende Kenntnisse hat über Zusammenhänge, und sich im Notfall auch weiterhelfen kann, und dem es wirklich nicht schadet, mal Mist zu bauen, oder sich sein System zu zerballern, **nur der sollte zu einer Beta-Version greifen.**

Das ist offensichtlich und verständlicherweise **hier im ST nicht** der Fall - bis auf ein paar Ausnahmen.

Also 'Oma/Opa',

erst mal Finger weg von einer Beta-Version.