

Endlose Stufen

Sicherheitslücken, Bugs und technischer Fortschritt bringen Distributionen jeden Tag in Bewegung. Admins, die fürs Deployment der Pakete die Verantwortung tragen, bleibt nichts als testen, testen, testen ... Markus Feilner



Wer die Qualität neuer Software bereits vor dem Einspielen (dem Rollout) auf Servern oder Clients testen muss (Quality Assurance, QA), sieht sich schnell einer recht komplexen Materie gegenüber. Experten empfehlen, die Software in mehreren Stufen (Stages) zu prüfen und sicherzustellen, dass jederzeit ein Zurück (Backout) zum Stand vor der Migration möglich ist.

Stages, Backout, Rollout, Rollback und Quality Assurance: Das Linux-Magazin hat fünf Experten gebeten, ihre Software-Teststrategien zu offenbaren. Zwei Dienstleister (B1 Systems und die LIS AG, zwei Distributoren (Univention und Red Hat) sowie der Test-Analyst des Münchner Linux-Projekts (Kasten „Limux im Test“) standen Rede und Antwort.

Große Einigkeit herrscht in einem Punkt: Essenziell ist es, neben der Produktionsumgebung eine Test- und Integrationsumgebung vorzuhalten. Dort lässt sich nicht nur feststellen, ob eine Softwarekombination auch nach dem Update einer Kompo-

nente läuft, sondern auch überprüfen, ob der neue Release-Stand alle Workflows der Firma weiterhin erlaubt.

Ralf Lang von B1 Systems (Abbildung 1) kann ein Lied davon singen: „Immer wieder entfernen Entwickler bei Versionsupdates Features oder verändern Plugin-Schnittstellen, die dann nicht mehr kompatibel sind. Viele Testszenarien lassen sich zwar mit Ant-Skripten [1] oder mit Jenkins [2] automatisieren. Aber der Rollout selbst erfordert detaillierte Vorausplanung, zu der auch das Abschätzen des nötigen Zeitfensters und das Entwickeln eines Rollback- oder Backout-Szenarios gehört, falls etwas schief läuft.“

Wartungsfenster, Rollback und Stages

Gelegentlich lösen Updates auch bisher völlig unbekannte Fehler aus. „Deshalb muss die IT-Abteilung das Wartungsfenster ausreichend groß planen, damit sie an seinem Ende ein funktionierendes System

– alt oder neu – garantieren kann. In größeren Umgebungen bietet es sich an, mit Pilotsystemen oder -anwendern die neue Software zu testen.“

In ähnlicher Richtung argumentiert Supportleiter Jürgen Schulz von der LIS AG: Stets „einen Schritt voraus“ ist für ihn ein Unternehmen, das auf den Rechnern seiner Admins die neueste Softwaregeneration verwendet, bevor es diese in die Produktivumgebung loslässt, sowohl auf dem Desktop als auch bei Serverprogrammen. „In unserem Staging erhalten die Arbeitsplätze der Mitarbeiter nicht immer automatisch alle Updates, so haben die Admins die Möglichkeit, frühzeitig Fehler und Probleme zu identifizieren und zu beheben“, erklärt Schulz.

Dabei ist jedoch viel organisatorische Vorarbeit zu leisten: „Die Operationen Guides und Workflows müssen klar definiert, schriftlich festgehalten und regelmäßig auf ihre Gültigkeit geprüft werden. Nur so lässt sich sicherstellen, dass die Herangehensweisen immer gleich sind. Das Vereinheitlichen der Systeme und der Infrastruktur muss das Ziel sein.“



Abbildung 1: Ralf Lang ist Linux Consultant und Developer bei B1 Systems.

Einig sind sich die Spezialisten auch darin, dass Admins in Firmen immer danach trachten, die „Komplexität gering zu halten und standardisierte Verfahren und APIs zu verwenden. Nicht zaubern oder tricksen, das kann bei Updates ein böses Erwachen geben, wenn die Software in einem Update die Arbeitsweise ändert. Dann beginnt alles von vorn.“
 Das von Schulz beschriebene Staging-Konzept hat zwei, idealerweise sogar drei Stufen: „Stage 0 ist ‚Leading Edge‘, also die neueste Softwareversion. Auf Stage 1 arbeiten die Tester mit einem von Stufe 0 abgeleiteten, bereits eingefrorenen Zustand. Stufe 2 sind die Produktivsysteme, die erst die Software bekommen, wenn die Tester grünes Licht geben. Dann wird Stage 1 zu 2 und Stage 0 zu 1 – und das Testing beginnt von vorn, im Idealfall als kontinuierlicher Prozess.“

Red Hat bietet Produkte

Auch Red Hat schwört auf ein gut geplantes Staging, sieht das Ganze aber auch stark in Organisationsstrukturen eingebettet (Abbildung 2): „Neben dem Einteilen der Abschnitte des jeweiligen Softwarelebenszyklus-Schritts (Stage) in Entwicklung (Dev), Qualitätssicherung (QA) und Produktion (Prod) ist es wichtig, Übergänge und Rollen zu definieren. Release and Deployment Management,

Change Management und Service Validation and Testing sind die Schlagworte, mit denen sich IT-Leiter auseinandersetzen müssen“, erklärt Dirk Hermann (Abbildung 3, Senior Solution Architect bei Red Hat). „Funktionalitäts-, Integrations-, Last-, Sicherheits- sowie Endanwender-Akzeptanz-Tests müssen die vier Szenarien (Install, Upgrade, Downgrade, Remove) sowie den stets mit zu testenden Backoutplan berücksichtigen.“
 Der Nachteil an all diesen Vorgaben ist offensichtlich: Admins und Tester müssen die Arbeitsweise ihrer User sehr genau kennen, weil sonst für die Kollegen elementare Funktionen eventuell unter den Tisch fallen. Jürgen Schulz: „Sehr wichtig ist es, Testsznarien zu entwickeln, in denen wichtige Abläufe und Keyfunktionen des Unternehmens vorkommen. Auf einer Testinstanz validiert ein Admin, ob das jeweilige Kriterium auch mit der neuen Software funktioniert. Dazu braucht es aber klare Vorgaben und exakte Definitionen.“
 Hinzu kommt, dass eine Enterprise-Distribution wie RHEL oder SLES zwar zertifizierte Middleware anbietet, manche Software aber zusätzliche Komponenten für den Betrieb braucht. „Da ist die IT-Abteilung selbst gefragt, Paketieren ist angesagt“, berichtet Ralf Lang. Für ihn ist ein typisches Beispiel „die Kombination aus PHP und Memcache: Die Distribution

bietet PHP und Memcache, aber die PHP-Memcache-Erweiterung selbst muss ein Dienstleister wie wir pflegen. Hilfreich für uns ist ein eigener, interner Buildservice wie Suses OBS [3], der Pakete für verschiedene Distributionen und Architekturen baut.“

Binär-Kompatibilität

Über die erwähnten Garantien der Konkurrenz hinaus hat Red Hat eine weitere wichtige Zusage im Angebot: „Unsere KABI/API-Garantie [4] verspricht hierbei die Stabilität der Schnittstelle zwischen dem Betriebssystem und dem darüberliegenden Binary oder Programming-Interface. Dies beinhaltet die Stabilität über Minor Releases hinweg sowie Stabilität und Unabhängigkeit von der darunterliegenden Hardware- oder Virtualisierungsplattform, also Bare Metal, VMware, Red Hat Enterprise Virtualization oder Microsoft Hyper-V.“
 Offensichtlich versuchen die roten Hüte hier, einen der größten Vorteile von Microsoft aufzugreifen: Auf den Redmonder Systemen bekommt der findige Anwender auch heute noch uralte Dos-Programme zum Laufen, während bei Linux die Abwärtskompatibilität in vielen Fällen hinterherhinkt.
 Moritz Mühlenhoff, Debian-Entwickler und beim Bremer Distributor Univenton für die Security-Maintenance des Univen-

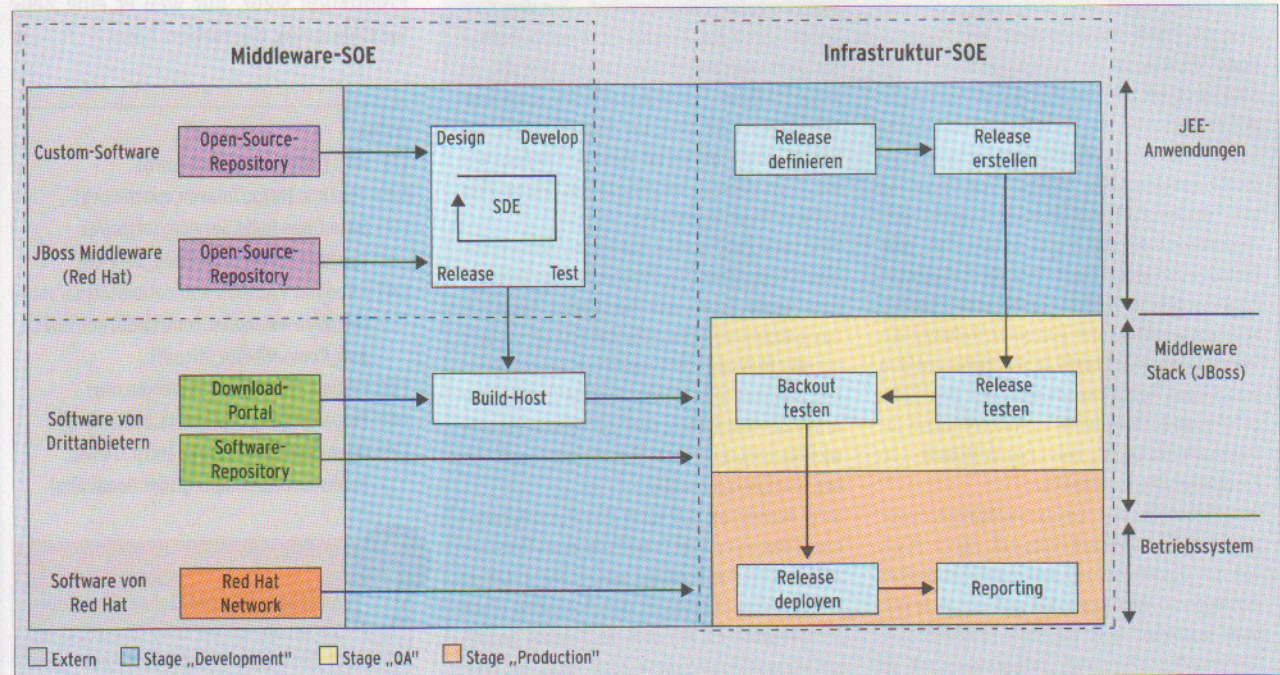


Abbildung 2: Reichlich komplex wirkt Red Hats Vorstellung einer Test-Infrastruktur für große Enterprise-Kunden im Standard Operating Environment (SOE).

tion Corporate Server zuständig, spricht sich für standardisierte Verfahren aus – und die Kombination mit der privaten Cloud. Schon die Testumgebung lasse sich ja ideal in einer Virtualisierung betreiben. „Dabei muss nicht die komplette Firmen-IT reproduziert werden, das lässt sich auch auf relevante und unerlässliche Dienste beschränken, zum Beispiel das LDAP-Verzeichnis.“

Univention: Ucs-Test

Univention verwendet ein selbst geschriebenes Framework namens Ucs-Test, das über Jenkins regelmäßig die Codebasis der Distribution überprüft. Die aus Debian importierten Pakete testen die Bremer darüber hinaus mit Tools wie Lintian [5] oder Piuparts [6]. „Für die Fachanwendungen vor Ort empfiehlt sich die enge Zusammenarbeit von Kunden, Dienstleitet und Anwender, bei kritischen Systemen eine Testfreigabe nach dem Vier-Augen-Prinzip“, erklärt Mühlenhoff.

Und er nennt noch einen menschlichen Aspekt, den OSI-Layer 8 sozusagen. Wer Desktopumgebungen auf neue Releasestände bringt, stolpert oft über Veränderungen in Usability oder Funktionsweise der GUIs, die für Benutzer in mehrerer



Abbildung 3: Dirk Herrmann, Senior Solution Architect mit Schwerpunkt System Management, Prozessintegration und Cloud bei Red Hat.

Hinsicht erklärungsbedürftig sind. „Hier helfen Pilotgruppen zu entscheiden, in welchem Umfang zusätzliche Dokumentation, Schulungen oder andere Unterstützungsmaßnahmen notwendig sind. Oft bedarf es auch individueller Erweiterungen, die wir über die Univention Management Console (UMC) in die Benutzerverwaltung integrieren.“

Ganz anders ist die Situation bei Firmen mit eigener Software-Entwicklung, die unter Red-Hat-Kunden häufiger zu finden sind: „Für diese Umgebungen deckt Continuous Integration neben der automatisiert erzeugten Releases auch einen umfangreichen und wertvollen Teil des Qualitätsmanagements ab“, meint Dirk

Herrmann. „In der Java-Software-Entwicklung haben sich Maven, Ant, Jenkins und Nexus etabliert, erweitert um Tools wie Sonar oder das Framework for Integrated Test (FIT).“

Viele der unter [7] gelisteten Werkzeuge lassen sich auch für Software-Entwicklung jenseits von Java nutzen. Wichtig ist, dass der Output der CI-Toolchain gleichzeitig den Input für ein zentrales Software Repository bildet, eine Definitive Software Library. Einen parallelen Stream für selbst erstellte Software sollten Admins in jedem Fall vermeiden“, empfiehlt Herrmann.

Viele Stolpersteine

Rollout Testing sinnvoll zu implementieren erfordert offenbar Weitsicht und Planung. Wer da nicht von vornherein sowohl die IT-Abteilung als auch die organisatorische Seite einbezieht, kann sich bei Updates auf Probleme gefasst machen. Auf Servern und mit virtuellen Desktop-Infrastrukturen lassen sich viele Testabläufe automatisieren, und der Trend, mehr und mehr Anwendungen ins Web zu bringen, macht in Zukunft für den Admin sicher vieles einfacher. Trotzdem muss er dafür sorgen, dass seine IT-Abteilung fortwährend den Stand kommender Softwareversionen testet. Sonst riskiert er, dass eines Tages die gesamte Produktion steht, nur weil er eine Zeile im Changelog übersehen hat. ■

Linux im Test

Im Linux-Projekt der öffentlichen Verwaltung der Stadt München stellt ein mehrstufiges Testkonzept die Funktionsfähigkeit der Komponenten mit spezifischen Anpassungen sicher, berichtet der dafür in der bayrischen Landeshauptstadt München verantwortliche Stefan Koehler: „Eine separate Testumgebung und -mannschaft, führt unter Produktivbedingungen Tests durch und stellt so vor dem Rollout einer jeden Major-Release die korrekte Funktion des Linux-Basisclients sicher. Danach kommen innerhalb der einzelnen Referate die Systemintegrationstests an die Reihe, wo den vielfältigen Fachanwendungen auf den Zahn gefühlt wird.“ Weil dafür im Referat lokale Infrastruktur nötig ist, muss das in München im Produktivsystem geschehen.

Die Strategie des Linux-Qualitätsmanagements in der Landeshauptstadt hängt stark von der Umgebung und den jeweiligen Anforderungen ab. Im Projekt hat sich ein mehrstufiger Testablauf über die Stufen Entwicklung, Systemintegrationstest, Integrationstest in den Referaten oder Fachabteilungen bis zum

User-Acceptance-Test bewährt. „Vor dem Rollout des Linux-Basisclients ist in jedem Referat der Administratoren Handarbeit gefragt, da immer wieder referatspezifische Anpassungen erforderlich sind. Die Aktualisierung von Serverkomponenten ist zwar kritischer, was die Auswirkungen angeht, in der Sache aber weitaus weniger komplex.“

Das Linux-Projekt setzt überall, wo es irgendwie möglich ist, Standardkomponenten ein: Open Office, Ubuntu, Firefox und Thunderbird. Diese Produkte testen wir nicht mehr umfassend, sondern nur die bisweilen umfangreichen Anpassungen und Bugfixes. Für uns ist also die Qualität der Arbeit des Distributors oder der Community sehr wichtig. Lösungen für Fehler, die wir finden, geben wir upstream weiter.

Sehr wichtig ist uns auch das Patchmanagement, das es erlaubt, innerhalb eines Releasezyklus kleinere Bugfixes auszuliefern. Linux bietet seinen Admins in den Referaten einen Mechanismus an, mit dem sie Patches und Fixes auswählen und diese auch vorab auf Test- und Produktivrechnern installieren können.“

Infos

- [1] Ant: [<http://ant.apache.org>]
- [2] Jenkins: [<http://maven.apache.org>]
- [3] Suse Open Build Service: [http://de.opensuse.org/Portal:Build_Service]
- [4] : Red Hat KABI/API Kompatibilitätsgarantie: [http://www.redhat.com/f/pdf/rhel/RHEL6_App_Compatibility_WP.pdf]
- [5] Lintian: [<http://lintian.debian.org>]
- [6] Piuparts: [<http://piuparts.debian.org>]
- [7] Open-Source-Testtools: [<http://www.opensourcetesting.org/functional.php>]



Linux Magazin Online PLUS

Auf Linux-Magazin Online Plus finden Sie einen Artikel über Open QA, ein Testtool für Distributionen und Appliances, das Suse entwickelt hat.