SolMan

ChaRMante Erweiterungen

Häufig werde ich in der Beratung darauf angesprochen, dass das Change Request Management im SolMan zunächst im Standard zum Einsatz kommen soll. Befassen sich IT-Verantwortliche mit den Konsequenzen für ihre IT-Prozesse, so entsteht schnell eine Wunschliste kundenspezifischer Erweiterungen.

Von Matthias Kneissl

Sprechen wir mit Kunden über das Change Request Management (ChaRM), so stellen wir immer als wesentliche Komponente heraus, dass es zwei wesentliche Arten von sogenannten Vorgangsarten gibt, die bereits im Auslieferungscustomizing enthalten sind: die „Normale Korrektur" und die „Dringende Korrektur". Beide Workflows bilden das Management von Softwareänderungen ab und sind bestens in das Transportwesen integriert. Der wesentliche Unterschied zwischen beiden Arten: Normale Änderungen können nur als Release produktiv gesetzt werden. Dringende Korrekturen, eigentlich von SAP gedacht für zeitkritische, umgehende Anpassungen, können zu jedem Zeitpunkt ohne Releasezuordnung produktiv gesetzt werden. In einer initialen Auslieferung sind diese beiden „Workflows" im System vorhanden und können kundenspezifisch noch angepasst werden, die Form der Produktivsetzung (als Release oder einzeln) ist jedoch vom System fest vorgegeben. Viele Kunden können mit der relaseorientierten Sichtweise nicht leben. Dies liegt einerseits daran, dass sich IT-Verantwortliche nicht mit dem Fachbereich anlegen und nur noch zu bestimmten Zeiten Funktionen produktiv setzen wollen. Zum anderen wird die Ausrede gebraucht „Wir sind agil und können uns daher nicht an ein Release binden". Daher setzt die Mehrheit der ChaRM-Anwender ausschließlich die Dringende Korrektur ein. In Umsetzungsprojekten existiert dann jedoch mehr als ein Workflow, sodass diese Vorgangsart in verschiedenen Varianten im System abgebildet wird – z. B. als Preapproved Change und als Major Change. Der Kontext dahinter bleibt immer die Dringende Korrektur und die damit verbundene Chance, die Changes einzeln produktiv zu setzen.

Stellt man IT-Verantwortlichen die Normale Korrektur und deren Softwareablauf vor, so stellt sich meist ein wesentlicher Vorteil gegenüber der Dringenden Korrektur dar: Es wird mit Transporten von Kopien gearbeitet. Da Entwickler meist nicht über die Testdaten auf dem Entwicklungssystem verfügen, die für einen umfangreicheren Test relevant sind, müssen die Änderungen in das Qualitätssicherungssystem gebracht und dort getestet werden. Im klassischen Kontext bedeutet dies, der Transport muss freigegeben und importiert werden. Dies ist an und für sich kein Problem, die meisten Unternehmen haben das Berechtigungswesen derart abgebildet, dass Transporte durch Entwickler freigegeben werden dürfen, und einen automatischen Import für die Importqueue des Qualitätssicherungssystems definiert.

In der Praxis führt es jedoch oft dazu, dass entwickelt, getestet, entwickelt und dann wieder getestet wird. Ich habe für Anforderungen schon derartige Zyklen gesehen, in denen mehr als hundert Transporte erzeugt wurden. In der klassischen Nutzung einer Dringenden Korrektur kann dies zwar über den Workflow abgebildet werden und auch der Überholerschutz tut sein Übriges, dennoch werden Produktivsetzungen solcher Changes umfangreich.

Mit Transporten von Kopien bleibt die Importqueue schlank, da die Objekte mittels der Auftragsart „Transport von Kopien" zwar ins Nachfolgesystem gestellt werden, diese „Versuche" aber nie in der Importqueue des Produktivsystems landen. In so einem Kontext sind mehr als hundert Versuche kein Problem. Ein wichtiges Feature ist dabei zu beachten: Der originale Transport bleibt bis zum Abschluss der Entwicklungstätigkeiten offen und die Objekte sind dann damit fest für diesen Change gesperrt. Dies hat den Vorteil, dass auch Überholer nicht mehr auftreten können. Mit geschickten Handgriffen und etwas Coding lässt sich diese Vorgehensweise auf die Dringende Korrektur übertragen. Wir stellen darüber hinaus den Workflow meist so ein, dass der originale Transport erst nach Abschluss der Tests durch den Key-User freigegeben wird. In diesem Workflow gehören dann Überholer vollkommen der Vergangenheit an. Gerade in Umfeldern, in denen nur mit der Dringenden Korrektur gearbeitet wird, möchte ja der Key-User nach Abschluss der Tests auch umgehend seine Änderungen im Produktivsystem sehen. Da das zeitliche Delta zwischen Freigabe der Objekte auf dem Entwicklungssystem und Import in das Produktivsystem gering ist (meist weniger als ein Tag), existiert auch keine Überholerproblematik mehr.

Matthias Kneissl ist einer der führenden SolMan-Experten in der deutschsprachigen SAP-Community.

Der **SAP Solution Manager** – oder SolMan, wie er liebevoll von der SAP-Community genannt wird – ist der zentrale Punkt für Service und Wartung. In einer hybriden IT-Architektur – on premise und on demand – gewinnt Maintenance nochmals an Bedeutung.

Bitte beachten Sie auch den Community-Info-Eintrag ab Seite 115