

Mit abat die Qualität bei ABAP-Entwicklungen sichern

“

Dieses Whitepaper gibt einen Überblick über die Standardprüfungswerkzeuge und ihre Konfigurationsmöglichkeiten, mögliche Integrationsszenarien und Genehmigungsprozesse mit diesen Werkzeugen werden vorgestellt und weiterführende QS-Prozesse diskutiert.



AUTOR

Jens Hollwedel
Senior Consultant

jens.hollwedel@abat.de

Zusammenfassung

Bei der Entwicklung von Kundenindividualsoftware ist ein begleitender Einsatz von qualitätssichernden Maßnahmen und Prozessen hilfreich, um während des Lebenszyklus das Ticketaufkommen zu begrenzen und eine schnelle Ergänzung oder Anpassung auf sich ändernde Bedürfnisse zu ermöglichen.

Zur Gewährleistung eines schnellen Return on Investment, müssen Aspekte bezüglich Sicherheit, Stabilität und Wartbarkeit in allen Phasen des Entwicklungsprozesses berücksichtigt werden. Die Werkzeuge und Prozesse der Qualitätssicherung (QS) sollten daher auch in allen Phasen integriert werden.

Dabei sind klug konfigurierte Prüfwerkzeuge verpflichtend, um häufige Probleme zu identifizieren und die Einhaltung der Entwicklungsrichtlinien zu kontrollieren. Prozesse zur Genehmigung von kritischen Befunden sollten etabliert und die jeweiligen Entwicklungen im Vier-Augen-Prinzip durch Code-Reviews abgenommen werden.

Durch die Integration der Qualitätssicherung in alle Phasen des Produktentstehungsprozesses

- wird das Ticketaufkommen im Produktlebenszyklus gesenkt,
- ist die Anpassung auf sich ändernde Anforderungen und Bedürfnisse schneller möglich,
- können kürzere Release-Zyklen eingeplant werden,
- kann die Integration neuer Technologien einfacher erfolgen.

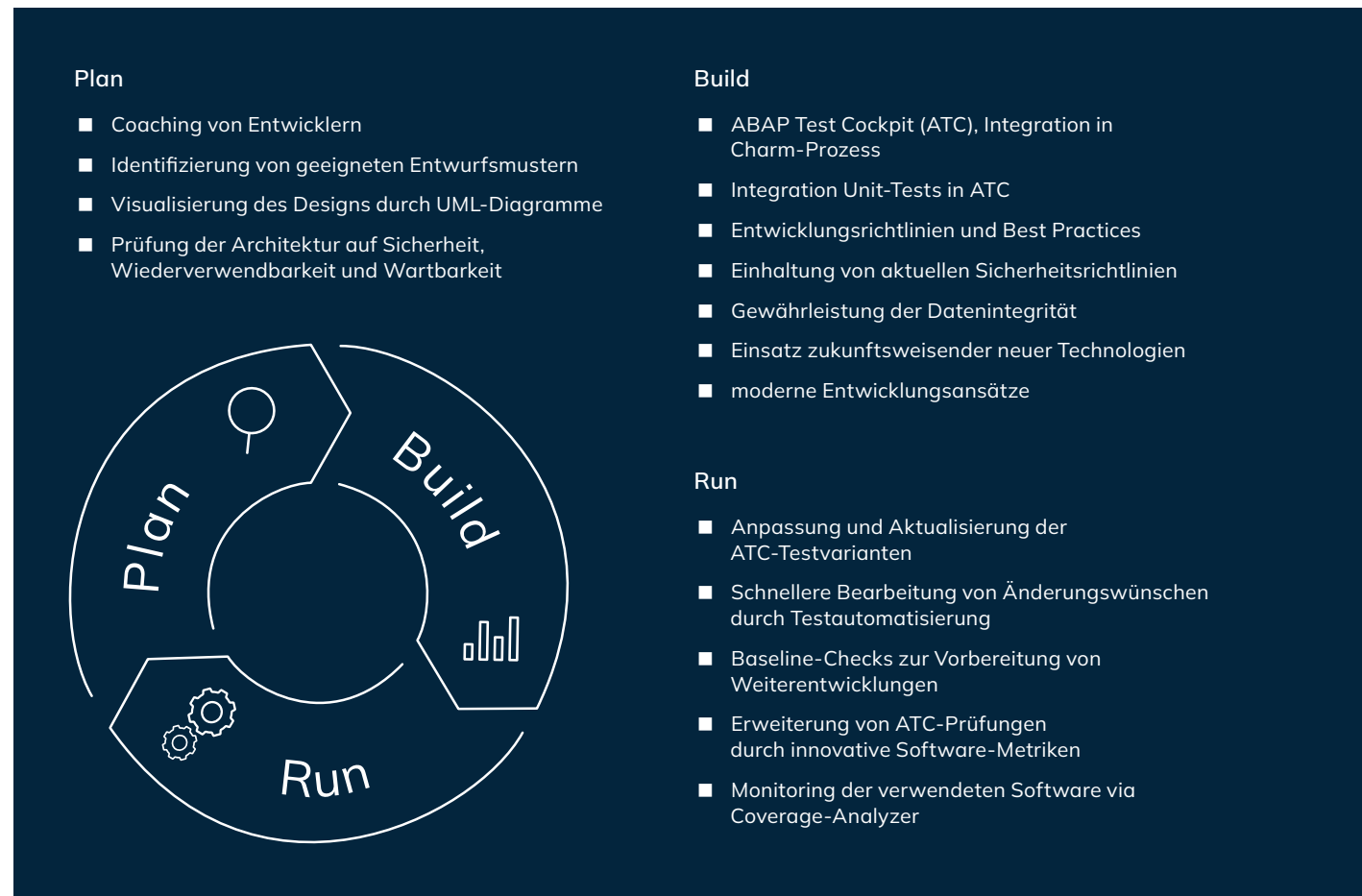


Abb. 1 Integration der Qualitätssicherung in unterschiedlichen Phasen des Produktentstehungsprozess

Vorbemerkung

In dynamischen Entwicklungsprojekten ist der ständige Wandel eine stetige Herausforderung. Werkzeuge und Methoden des Qualitätsmanagements sollen dabei helfen, diesen Wandel zu kontrollieren und Leistungsfähigkeit, Sicherheit, Wiederverwendbarkeit sowie Wartbarkeit der Software zu gewährleisten. Das bedeutet kürzere Entwicklungszeiten, schnellere Tests, weniger Nacharbeit in den Entwicklungsprojekten und erhöht somit den Return on Investment.

Qualitätsmanagement in der ABAP-Entwicklung

Nachhaltige Softwareentwicklung zeichnet sich dadurch aus, dass der gesamte Produktlebenszyklus ganzheitlich durch das Qualitätsmanagement betrachtet wird. Dies beginnt in der konzeptionellen Phase, in der geeignete Entwurfsmuster als Grundlage des Designs dienen und wiederverwendbare Algorithmen als Module konzipiert werden. Der Entwicklungsprozess wird sowohl durch werkzeuggestützte Codeprüfungen als auch durch Code-Reviews begleitet.

Eine ganzheitliche Sicht auf die Qualitätssicherung in der ABAP-Entwicklung umfasst mehrere Aspekte, von Entwicklungsrichtlinien über Prüfwerkzeuge bis hin zu Freigabeprozessen und manuellen Design- und Code-Reviews. Diese Elemente werden im Folgenden detailliert vorgestellt.

Die Qualitätssicherung der ABAP-Entwicklung beginnt beim konkreten Lösungsdesign und nur in besonderen Ausnahmefällen in der fachlichen Konzeptionsphase.

Akzeptanz von qualitätssichernden Maßnahmen

Für einen erfolgreichen Projekteinsatz von qualitätssichernden Maßnahmen bei der ABAP-Entwicklung ist es erfahrungsgemäß sehr wichtig, alle Projektmitglieder mit einzubeziehen. So sollte etwa die Rückmeldung der ABAP-Entwickler in die Ausgestaltung verwendeter Standardprüfvarianten berücksichtigt werden. Auch die Fachbereiche, die Fachberater und die Tester sollten entsprechende Zeiten für Qualitätssicherung und eventuell erforderliche Nacharbeiten von Anfang an einplanen.

Neben den entsprechenden QS-Prozessen und Prüfschritten sollten regelmäßige Abstimmungen zwischen der QS und den Entwicklungskoordinatoren erfolgen.

Entwicklungsrichtlinien

Ebenso, wie sich die Software verändert und an neue Anforderungen angepasst wird, müssen auch die Entwicklungsrichtlinien immer wieder an den sich wandelnden Stand der aktuellen Technologie angepasst werden. Idealerweise gehen dabei unterschiedliche Quellen in die Überarbeitung und kontinuierliche Verbesserung der Entwicklungsrichtlinien ein: Erkenntnisse und Erfahrungen aus den Entwicklungsprojekten, sowie Best-Practices aus Dokumenten der SAP und der DSAG.

Die Entwicklungsrichtlinien sollen dabei aber kein Entwicklerhandbuch ersetzen, sondern klare Verbote und Forderungen in Bezug auf die ABAP-Entwicklung liefern. Sie sollen Vorgaben zur Benennung von Repository-Objekten machen und sowohl

Design- als auch Implementierungsaspekte abdecken, wobei letztere möglichst statisch testbar sein sollen (was aber nicht immer möglich ist).

Neue Versionen der Entwicklungsrichtlinien sollten den Entwicklern vorgestellt werden, bevor sie die bisherige Version als verbindliche Richtlinie ablösen. Möglicherweise können die Richtlinien auch in Workshops vorgestellt werden, um Feedback zu sammeln und bei Bedarf einzuarbeiten.



Prüfwerkzeuge

Für die Qualitätssicherung der ABAP-Entwicklungen bietet die SAP verschiedene Werkzeuge an, die in das System integriert sind. Mittlerweile gibt es sogar Szenarien für die Prüfung älterer Systeme, mittels eines Remote-ATC-Systems.

Standardprüfwerkzeuge in der Systemlandschaft

Die von der SAP zur Verfügung gestellten Prüfwerkzeuge sind in die Entwicklungsumgebungen, also die ABAP-Workbench und Eclipse mit den ABAP Development Tools vollständig integriert.

Die erweiterte Syntaxprüfung (SLIN) ist das älteste der hier beschriebenen Instrumente, in die neueren Prüfwerkzeuge integriert, und muss dadurch nicht separat ausgeführt werden. Später dann wurde der SAP Code Inspector (SCI) eingesetzt, der in modernen QS-Szenarien noch immer für bestimmte Aspekte (siehe Prüfvarianten) verwendet wird. Darauf aufbauend wurde von der SAP das aktuelle Prüfwerkzeug, das ABAP Test Cockpit (ATC), bereitgestellt.

Für den SCI und darauf aufbauend das ATC bietet die SAP verschiedene vorkonfigurierte Prüfumfänge an, die in Form von Prüfvarianten zur Durchführung von SCI- bzw. ATC-Prüfungen eingeplant werden können. Daneben lassen sich aber auch eigene Prüfvarianten konfigurieren, die eine beliebige Menge von Standardprüfungen umfassen, die wiederum über Parameter an die in den Entwicklungsrichtlinien definierten Regeln angepasst und gegebenenfalls noch unterschiedlich priorisiert werden können.

Integration Prüfwerkzeuge

Alle Prüfwerkzeuge, also die erweiterte Syntaxprüfung, der SAP Code Inspector und das ABAP Test Cockpit können direkt aus der Entwicklungsumgebung aufgerufen werden, um beispielsweise eine gerade entwickelte oder abgeänderte Klasse zu prüfen.

Wenn der Entwickler dabei keine besondere Auswahl trifft, wird die im System eingestellte Standardprüfvariante genutzt. Der Entwickler kann aber auch je nach Bedarf andere Prüfvarianten auswählen.

Neben der Integration in die Entwicklungswerkzeuge können regelmäßige Prüfläufe auch als Hintergrundjobs laufen und dadurch beispielsweise neben Standardprüfungen auch spezielle Prüfumfänge eingeplant werden. Alle Ergebnisse der Standardprüfungen werden den jeweilig verantwortlichen Entwicklern in ihrem Arbeitsvorrat angezeigt.

Eine weitere Integrationsmöglichkeit besteht darin, Prüfungen in die Transportfreigabe zu integrieren (siehe Freigabe von Transportaufträgen und Genehmigungsworkflow), um sicherzustellen, dass alle Entwicklungen vor Import in Q-Systeme diese Prüfung durchlaufen haben und keinerlei ungenehmigte Prio-1-Befunde übrigbleiben.

Prüfvarianten

Die Pflege für die Prüfvarianten erfolgt sowohl für den SAP Code Inspector (SCI) als auch das ABAP Test Cockpit (ATC) in der SAP-Oberfläche des SCI.



Aus einer Menge von bereitgestellten vordefinierten Prüfvarianten lässt sich auswählen, welche Variante für welchen Prüflauf genutzt werden soll. Daneben lässt sich aber auch definieren, welche Variante als Standardprüfvariante fungieren soll. Für den SCI wird sie in Tabelle SCICHKV_ALTER¹ eingetragen, im ATC kann sie in den ATC-Einstellungen hinterlegt werden.

Es lassen sich im SCI aber auch eigene Prüfvarianten definieren. Dabei werden aus einer Menge von Prüfkategorien beliebige Prüfungen ausgewählt, die dieser Prüfvariante hinzugefügt werden sollen. Auch die Priorisierung der Prüfungen lässt sich dabei individuell in drei Prioritätsstufen anpassen. Einige Prüfungen können noch durch Eingabe weiterer Parameter gesteuert werden.

¹ oder man nennt die eigene Prüfvariante

Entwicklung eigener ATC-Prüfungen

Neben den vordefinierten Prüfungen lassen sich auch noch eigene Prüfungen ergänzen. Die SAP stellt dafür ein API bereit, so dass kundenindividuelle Prüfungen als Add-ons in die Prüfwerkzeuge einfach integriert werden können.

Erfahrungsgemäß lassen sich damit die Entwicklungsrichtlinien besser abdecken, wobei auch komplizierte Namenskonventionen und Pakethierarchien berücksichtigt oder unterschiedliche Konventionen für unterschiedliche Pakete gesteuert werden können.

Zusätzlich können Fehler oder Probleme, die häufig auftauchen², direkt mit eigenen Prüfungen ermittelt werden, woraus sich eine große Zeitersparnis beispielsweise bei manuellen Code-Reviews ergibt.

Einführung von verpflichtenden ATC-Prüfungen

Die Einführung regelmäßiger und verpflichtender ATC-Prüfungen erfolgt idealerweise in mehreren Schritten.

In einem ersten Schritt sollte eine neue spezifische Prüfvariante, die zuerst nur Standardprüfungen umfasst, allen Entwicklern zur freiwilligen Verwendung zur Verfügung gestellt werden – diese wird als Standardprüfvariante definiert. In gleicher Phase sollten die Rückmeldungen der Entwickler gesammelt werden, um die Prüfvariante entsprechend anzupassen, beispielsweise um Prüfungen hinzuzufügen oder Prioritäten von Befunden anzupassen.

Diese Prüfvariante sollte zwar in die Transportfreigabe³ integriert werden, aber zuerst lediglich bei Prio-1-Befunden informieren.

Nach dieser Einführungsphase werden regelmäßige Hintergrundprüfläufe eingeplant, allerdings zuerst noch ohne die E-Mail-Benachrichtigung von Verantwortlichen bei Prio-1-Befunden. Gleichzeitig sollten bei der Transportfreigabe Prio-1-Befunde nun als transportverhindernd gewertet werden, so dass die Freigabe (siehe Freigabe von Transportaufträgen und Genehmigungsworkflow) erst durch einen bestandenen Genehmigungsschritt erfolgen kann.

In einem letzten Schritt sollten dann bei den regelmäßigen Prüfläufen auch die verantwortlichen Entwickler bei Prio-1-Befunden per E-Mail informiert werden.

Parallel dazu sollte eine Testprüfvariante bereitgestellt werden, die weitere Prüfumfänge und dazu auch projektspezifische Add-on-Prüfungen umfasst. Diese Testprüfvariante kann dann wieder von den Entwicklern optional genutzt werden und wird nach einer Evaluierungsphase die neue verbindliche Prüfvariante.

abat Best-Practices für ATC-Prüfungen

- Eine projektspezifische Prüfvariante sollte als Standardprüfvariante mit angepassten Priorisierungen im System eingerichtet werden, um die Entwicklungsrichtlinien möglichst umfänglich abzudecken.
- Ein regelmäßiger ATC-Prüflauf mit der projektspezifischen Standardprüfvariante sollte einmal wöchentlich über das gesamte System erfolgen.
- Bei Prio-1-Befunden sollten die verantwortlichen Entwickler nach einer Anfangsphase per E-Mail automatisch aus dem System informiert werden.
- Weitere Prüfvarianten für regelmäßige Prüfläufe, die neben den Prüfungen der Standardvariante noch die Ermittlung von Kennzahlen umfassen, sollten nach einer Anfangsphase eingerichtet werden.
- In einer späteren Phase sollten angepasste Namenskonventions- und Paketprüfung, sowie eine Prüfung auf häufige Probleme in Code-Reviews als Add-on-Prüfungen hinzugefügt werden.

² z.B. eine fehlende Erweiterungskategorie bei Strukturen oder Datenbanktabellen

³ Der SCI-Testtreiber sollte deaktiviert, nur das ATC sollte eingebunden sein.



abat Best-Practices für ATC-Transportintegration

- Einrichtung einer QS-Mailbox
- Unterbindung von Transportfreigaben bei Prio-1-Befunden
- Ignorieren der Unterdrückung von Befunden
- Genehmigung von Exemptions bei der Transportfreigabe durch Qualitätsmanager (4-Augen-Prinzip)

Freigabe von Transportaufträgen und Genehmigungsworkflow

Neben der Integration der Prüfwerkzeuge, insbesondere des ATC in die Entwicklungsumgebung und der Durchführung von regelmäßigen Prüfläufen, sollte das ATC in die Freigabe von SAP Workbench-Aufträgen im Transportmanagement integriert werden.

Damit können beispielsweise Transportfreigaben verhindert werden, falls im ABAP-Code der dazugehörigen Workbench-Objekte noch Prio-1-Befunde vorhanden sind. An dieser Stelle werden idealerweise vom Entwickler gesetzte Pragmas oder Pseudokommentare nicht ohne weitere Prüfung genehmigt. Die individuelle Aussteuerung der Transportintegration des ATC kann durch ein entsprechendes BADl feingesteuert werden.

Falls also ein Transport mit einem Prio-1-Befund freigegeben werden soll, dann muss eine automatische Benachrichtigung aus dem System an eine QS-Mailbox erfolgen, auf die alle Qualitätsmanager Zugriff haben. Der Entwickler würde im ATC einen Genehmigungsantrag begründen, der dann durch einen Qualitätsmanager geprüft und bestenfalls genehmigt wird. Das ATC bietet zur Genehmigung von Exemptions und zur Kommunikation mit den Entwicklern schon eine funktionierende Umgebung.

Für dieses Quality-Gate, das Entwicklungen standardmäßig durchlaufen müssen, sollte aber auch eine Möglichkeit bedacht werden, um Notfalltransporte ohne Verzögerungen durchführen zu können.

Solution-Manager-Integration

Häufig wird der Solution-Manager zur Steuerung von Entwicklungen und Verwaltung der SAP-Transporte genutzt. In einem solchen Szenario können neben den zugehörigen SAP-Transporten auch die Ergebnisse der ATC-Prüfungen im Solution-Manager angezeigt und eingebunden werden.

Design- und Code-Reviews

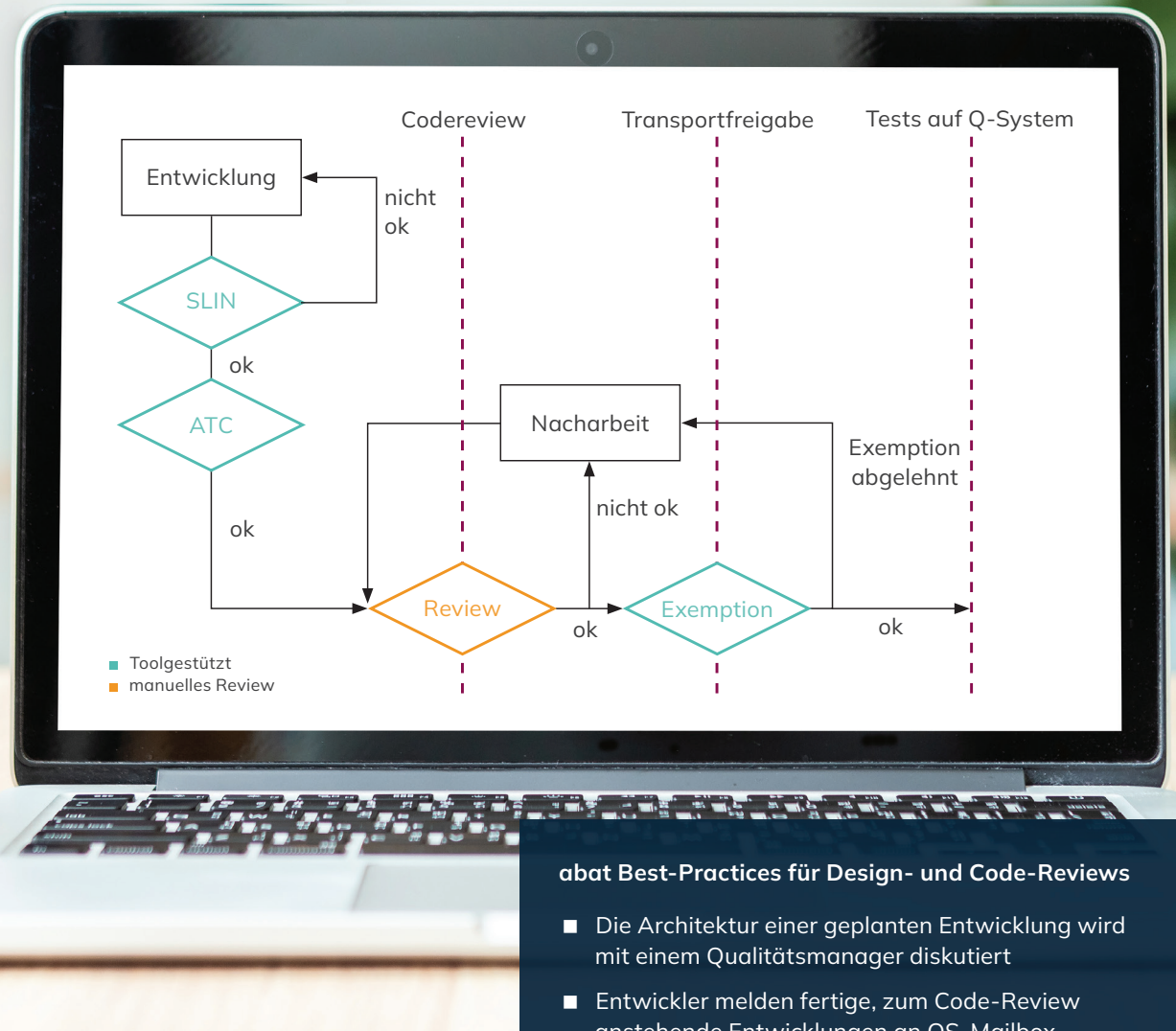
Über die automatisierten Prüfungen durch Werkzeuge zur statischen Code-Analyse hinaus, sollten Design- und Code-Reviews als feste Bestandteile des Softwareentwicklungsprozess etabliert werden. In einem agilen Projektvorgehen (bei anderen Projektmethodiken gilt entsprechendes) könnte der Design- und Code-Review-Schritt etwa folgendermaßen gestaltet werden:

In der agilen Umsetzungsphase gibt es in jedem Sprint zwei Zeitpunkte, an denen Qualitätssicherungsschritte sinnvoll sind: Im Sprint Planning werden User-Stories (also die agilen Konzepte), die die Kriterien der „DoR“ (Definition of Ready) erfüllen, zur Umsetzung eingeplant. Am Ende des Sprints, zum Sprint-Review, werden umgesetzte User-Stories, die die „DoD“ (Definition of Done) erfüllen, abgenommen.

Zu einer solchen Story sollte zuerst ein technisches Design, idealerweise in Form von UML-Diagrammen, erstellt werden. Dieses Design wird dann in einem Design-Review abgenommen. Erst im Anschluss daran erfolgt die Realisierung. Mit Werkzeugen, wie dem SAP Code Inspector und dem ABAP Test Cockpit, durch Unit-Tests und manuelle Code-Reviews wird geprüft, ob die fertige Story die DoD-Kriterien erfüllt. Der Code-Review-Schritt sollte durch eine 4-Augen-Prüfung des Quellcodes erfolgen. Die einzelnen Qualitätssicherungsschritte können entweder durch die Scrum-Teams selbst oder besser noch durch ein unabhängiges Qualitätssicherungsteam durchgeführt werden.

Wenn die Code-Reviews durch Qualitätsmanager erfolgen, die nicht Teil des Entwicklungsteams sind, dann sollten über eine QS-Mailbox Reviews angefordert werden können. Der Reviewer meldet dann eventuelle Befunde an den Entwickler und gibt

das Design der Entwicklung bzw. die Implementierung frei. Dazu dokumentiert er den Status beispielsweise in einem Projektmanagement-Werkzeuge, wie etwa JIRA oder dem SAP Solution Manager Add-on Focused Build.



abat Best-Practices für Design- und Code-Reviews

- Die Architektur einer geplanten Entwicklung wird mit einem Qualitätsmanager diskutiert
- Entwickler melden fertige, zum Code-Review anstehende Entwicklungen an QS-Mailbox
- Qualitätsmanager sendet Bericht mit gegebenenfalls nötigen Nacharbeiten an Entwickler
- Status wird in einem geeigneten Projektmanagement-Werkzeug dokumentiert
- Regelmäßige Jour-Fixes mit Lead-Entwicklern und Qualitätsmanagern finden statt

Altlasten, technische Schulden und ATC Baseline

In einigen Systemen kommt es über die Zeit zu sehr vielen ABAP-Test-Cockpit-Befunden. Diese können dann neu aufgetretene Befunde «maskieren», so dass kein Überblick über aktuell zu behebende Probleme besteht. Deswegen sollte für die Standardprüfung eine Baseline ermittelt werden, in der die Legacy-Findings aufgenommen werden und so aus der relevanten Menge der zu behebenden Befunde verschwinden. In diese Baseline sollte eventuell verbleibende Probleme nach einem Go-live ebenfalls aufgenommen werden.

Für die Behebung von problematischen Befunden in der Legacy-Software sollten Prüfläufe über die gesamte Code-Basis erfolgen, so dass die relevanten Objekte identifiziert werden können.

Der ABAP Call Monitor sollte auf dem Produktivsystem aktiviert werden, um nicht mehr benötigte Altprogramme zu identifizieren. Daneben sollte auch über den Zeitraum von wenigen Wochen eine Protokollierung der Datenbankzugriffe auf dem Produktivsystem mittels des SQL-Monitors erfolgen. Die so erhaltenen Ergebnisse lassen sich mit den Performance-Prüfungen des ABAP Test Cockpit kombinieren, um so besonders performance-kritische Objekte zu identifizieren.

ANSPRECHPARTNER

Jens Hollwedel

Senior Consultant

jens.hollwedel@abat.de

ÜBER UNS



Die abat Gruppe, 1998 gegründet, ist SAP-Dienstleister, innovativer Softwareentwickler und Anbieter von Komplettlösungen für die softwaregestützte Prozessoptimierung –

vor allem in den Kernbranchen Automotive und Diskrete Fertigung sowie in logistischen Prozessen und der Fertigungssteuerung. Mit unseren sechs Leistungsbereichen verschaffen wir Unternehmen den Freiraum, den sie für neue Ideen, effiziente Prozesse und zukunftsweisende Lösungen benötigen.

Im Leistungsbereich **consult** beraten und begleiten wir Sie in allen Phasen eines SAP-Projektes – von der Konzeption über die Implementierung bis zum Betrieb Ihres SAP-Systems. Dank abat **manufacture** erhalten Sie digitale Hochverfügbarkeitslösungen zur Produktionssteuerung im Bereich der komplexen Fertigungsindustrie. Mit abat **transform** bieten wir innovative und einzigartige Lösungen, die Sie besonders machen: von AI über Cloud bis zu X-Reality. Der Bereich **plm** hält übergreifende Prozessberatung bereit, mit dem Ziel, einen durchgängigen Datenfluss über PLM, ERP und MES zu erreichen. Angebote aus dem Bereich **protect** helfen Kunden dabei, Informationen zu schützen sowie Vertraulichkeit, Verfügbarkeit und Integrität in Geschäftsbeziehungen zu bewahren. Und schlussendlich beraten unsere Expert*innen des Bereichs **sustain** u.a. dazu, wie sich Nachhaltigkeit sowie die Berichterstattung darüber strategisch und strukturell im Unternehmen verankern lassen.

abat | An der Reeperbahn 10 | 28217 Bremen | +49 421 43 04 60 | www.abat.de