

## Anwendungsentwicklung optimieren

- Den Code verstehen
- Die Codequalität verbessern
- Die Produktivität erhöhen

# GUTES KANN NOCH BESSER WERDEN

Sowohl für Cobol- und Natural-, als auch für RPG-Entwickler gibt es keinen Grund mehr, in den Entwicklungsparadigmen der Vergangenheit zu verharren und damit mögliche Produktivitätschancen ungenutzt zu lassen. Auch für diese Legacy-Sprachen stehen heute Entwicklungsumgebungen auf Basis des Eclipse-Frameworks zur Verfügung, z.B. von IBM. Diese bieten den komfortablen Einstieg in die Alt-Sprachen in der gewohnten Arbeitsumgebung für junge Entwickler und vereinfachen gleichzeitig den erfahrenen Legacy-Entwicklern den Zugang zu Java. Dadurch kann der Generationswechsel in der Anwendungsentwicklung sicher erfolgen. Doch noch einen weiteren Vorteil bieten diese grafischen Entwicklungsumgebungen: Sie eröffnen - im Gegensatz zur bisherigen Zeichen- und Zeilen-Orientierung - alle Möglichkeiten der visuellen Aufbereitung des Codes. Und - last but not least - neue Möglichkeiten für einen modernen Entwicklungsprozess samt Versionskontrolle, Teamarbeit, Änderungsmanagement und Testunterstützung.

## DIE CODEQUALITÄT VERBESSERN

1. Geringere Kosten für die Wartung und Weiterentwicklung der Anwendung
2. Höhere Qualität der Auslieferungen bzgl. Stabilität und Performance
3. Kürzere Reaktionszeiten bei Änderungsanforderungen
4. Messbare Kriterien für die Wartbarkeit der Anwendung als Basis für einen kontinuierlichen Verbesserungsprozess

**Sind die genannten vier Punkte erstrebenswerte Ziele für Sie und Ihr Team?**

Die **Code-Quality-Engine** ermöglicht eine Verbesserung der Code-Qualität und stellt sicher, dass auch bei der Weiterentwicklung definierte Qualitäts-Regeln eingehalten werden. Diese Regeln

können entweder vom System vorgegeben oder kundenindividuell implementiert sein. Damit werden manuelle Fehlerquellen eliminiert lange bevor die Sourcen in die Testphase gelangen und enorme Kosten für die Fehlersuche und -behebung entstehen würden. Ebenso ist das Quality Plugin in die bestehende Entwicklungsumgebung perfekt integriert, so dass es keine Hürden für eine schnelle Einarbeitung gibt. Mit dem routinierten Einsatz des Quality Plugins reift schnell das Gefühl von Sicherheit und Erleichterung bei allen Beteiligten - Entwicklern, Projektleitern und dem Budgetverantwortlichen.

Die Ergebnisse sprechen für sich: Wenn der Arbeitsalltag an vielen kleinen Stellen optimiert wird und sich die Fehleranfälligkeit deutlich reduziert, dann machen sich Arbeitersparnis und Qualität schnell bezahlt.

# TECHNICAL INSIGHTS

Die Quality-Rules dienen zur Einhaltung von Programmier-Richtlinien bzw. QA-Standards, mit folgenden Zielen:

## 1. Die Lesbarkeit des Codes wird erhöht

Durch feste Vorgaben für Namenskonventionen, Größenbeschränkungen etc. reduziert sich der Aufwand für die Fehlersuche erheblich. Ebenso wird der Einarbeitungsaufwand für neue Mitarbeiter deutlich geringer

## 2. Die Wartbarkeit des Codes wird verbessert

Durch die Einschränkung der Nutzung gewisser Konstrukte, wie z.B. GOTO oder zu sehr verschachtelten Kontrollflüssen sowie der Erkennung unbenutzter Codestrecken (Dead Code) wird eine effizientere Fehlersuche und -behebung möglich und die qualitativ hochwertige Weiterentwicklung des Codes abgesichert

## 3. Die Fehleranfälligkeit des Codes wird reduziert

Kritischer Konstrukte wie typfremde Zuweisungen, typfremde Parameterlisten, nicht-initialisierte Datenelemente, fehlendes Abprüfen von Fehlercodes etc. werden implizit überwacht und damit verhindert. Durch möglichst weitgehendes und vorausschauendes Vermeiden von Fehlersituationen, die sich erst zur Laufzeit materialisieren, werden die Kosten für deren Analyse und Behebung im Test- und Produktivbetrieb komplett verhindert

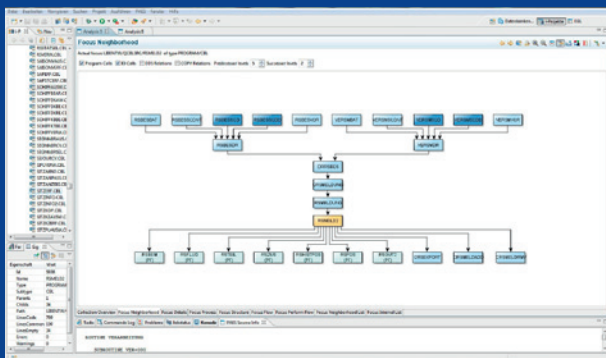
## 4. Die Performance des Codes kann verbessert werden

Die Einschränkung von Algorithmen, die erfahrungsgemäß zu Engpässen führen (Öffnen von Cursors innerhalb von Schleifen, Select \* etc.) kann über Regeln verhindert werden. Auch diese problematischen Situationen werden sonst erst zur Laufzeit sichtbar und verursachen Kosten und Aufwand in der Analyse und Behebung.

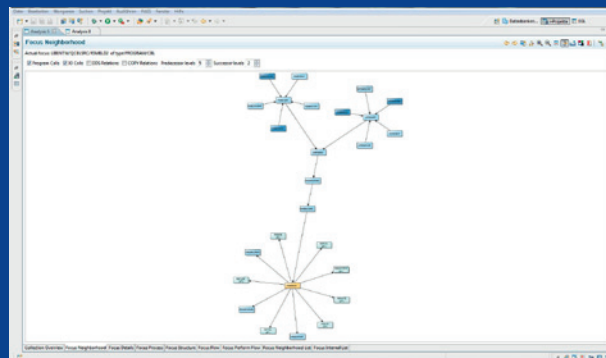
## FEATURES & FUNCTIONS

- Für jede Regel kann einzeln die Severity und die Berücksichtigung für den Commit-Vorgang festgelegt werden
- Jeder Benutzer kann separat festlegen, welche Meldungen er überhaupt sehen will
- Jeder Benutzer kann separat festlegen, welche Severity-Levels er als Annotations im Sourcecode sehen will (Markierungen)
- Einzelne Regeln können über Parameter gesteuert werden, z.B. die Suchtiefe bei Verschachtelungen
- Anzeige der Meldungen in TreeView mit frei wählbarer Gruppierung oder in Form eines Grids
- Persönliche Filter zu jedem Zeitpunkt (de)aktivierbar
- Anzeige der Meldungen ohne erneutes Parsen möglich
- Anzeige von Statistikübersichten, frei gruppierbar
- Verlinkung der Meldungen zu den einzelnen Codestellen inkl. Annotations
- Exportmöglichkeiten nach Excel (csv-Format)
- Direkte Integration in den Subversion-Commit-Vorgang

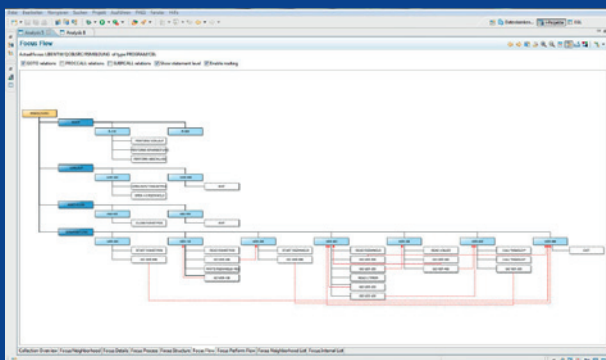
# DEN CODE VERSTEHEN – LEICHTER GESAGT ALS GETAN



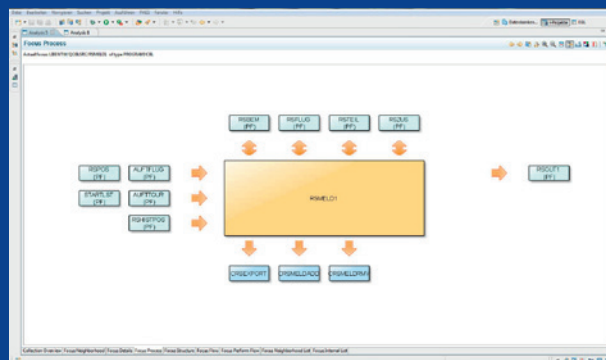
Kundenzitat: „Für so eine Übersicht saß ich bisher einige Stunden am Rechner - genial, dass das auf Knopfdruck online da ist.“



Kundenzitat: „ Mit solchen Sichten fällt es viel leichter die richtigen Schnitte für die Überführung unserer Anwendung in eine service-orientierte Architektur zu überführen.“



Kundenzitat: „Nur durch die Feldverfolgung auf Basis PAAS waren wir in der Lage, alle Stellen effizient ausfindig zu machen, die von der Feldlängenänderung eines Primärschlüsselfeldes der Anwendung betroffen waren. Das hat uns enorme Testaufwände und manuelle Arbeiten gespart.“



## DEN CODE VERSTEHEN

Sinkende IT Budgets zwingen die Entwicklungsabteilungen mit weniger Personal den gleichen oder sogar einen höheren Arbeitsaufwand zu bewältigen. Deshalb muss die Entwicklungsarbeit so schlank und pragmatisch wie möglich sein. Unrunde Entwicklungsprozesse mit zahlreichen manuellen Arbeitsschritten verlängern unnötig die Entwicklungszeiten.

Gleichzeit leben gerade Bestandsanwendungen häufig mehr als gedacht, sie werden weiterentwickelt und müssen bei gesetzlichen oder fachlichen Änderungen angepasst werden. Häufig fällt dann auf, dass es mit einer guten Dokumentation der Systeme nicht all zu weit her ist oder dass benötigte Aufwandsabschätzungen reines „Bauchgefühl“ sind. Um hier auf einer validen Datenbasis unterwegs zu sein, bieten sich die **Analyse- und Impact-Plugins** von PAAS an.

Der Entwickler kann damit aus der gewohnten Arbeitsumgebung heraus sowohl seine eigenen als auch die Sources anderer Teammitglieder sehr rasch analysieren, Abhängigkeiten verstehen und den Datenfluss nachvollziehen. Das zugehörige datenbankbasierte Repository sowie umfangreiche Symboltabellen bieten zudem die Grundlage für die Durchführung von Codeänderungen auf Feldebene und unterstützen bei Massenänderungen.

## VORGEHEN:

1. Wir laden die von Ihnen bereitgestellten Sources in das PAAS-Repository
2. Wir generieren die zugehörigen Meta-Daten und richten die gewünschten Engines ein
3. Dann starten wir in einen kurzitägigen Workshop mit Ihrem Team und stellen die Ergebnisse vor - Sie erhalten eine spannende Sicht und viele neue Erkenntnisse über Ihre Sources - eine interessante Reise kann beginnen.



### ANWENDUNGSENTWICKLUNG OPTIMIEREN

- Eclipse basierte IDEs (RDz, RDP)
- Application understanding
- Codequalität verbessern



### USERINTERFACES ERNEuern

- Web 2.0 Integration + Erweiterung
- Unicode Enabling
- Mobile device Anbindung über WebServices

## ADABAS Natural

### ADABAS / NATURAL MIGRATION

- Adabas nach RDBMS Migration
- Natural und Java Koexistenz
- Web 2.0 für Natural MAPs



### ANWENDUNGS- TRANSFORMATION

- Konsolidierung von 4GL nach Cobol
- zIIP Enablement – von Cobol nach Java
- maßgeschneiderte Transformationslösung



Heidi Schmidt  
Geschäftsführerin  
Vertrieb+Marketing  
Tel. +49 751/56140-229  
schmidt@pks.de



Roland Zurawka  
Geschäftsführer  
Produkte+Services  
Tel. +49 751/56140-222  
zurawka@pks.de



**PKS Software GmbH**  
Georgstraße 15  
88214 Ravensburg  
GERMANY

Telefon: +49 (0) 751 5 61 40-0  
Telefax: +49 (0) 751 5 61 40-500

[www.pks.de](http://www.pks.de)