Qualitätssicherung durch automatisierte Codeanalyse

scch { }

Ein grober Fehler in der Programmierung wird schnell transparent. Ein kleiner Fehler in der Programmierung von Steuerungen kann sich sehr lange verstecken und dennoch einen enormen Schaden anrichten.

Ziel ist es, den Code automatisch nach Fehlern zu prüfen, aber beispielsweise auch die Einhaltung von (internen) Programmierregeln zu kontrollieren.

Ein langwieriges Fehlersuchen wird damit reduziert und die Qualität beim Kunden gesteigert.

Das Projekt ist für die Einreichung in einer Förderschiene geplant, bei der bis zu 65% der Kosten für die Entwicklung übernommen werden (Collective Research).

Forschung

JKU JOHANNES KEPLER UNIVERSITÄT LINZ

scch { }

Projektmanagement

MC MECHATRONIK CLUSTER

Unternehmen

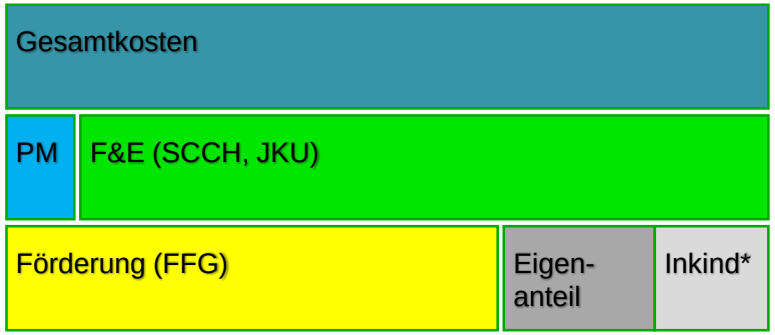
Nutzen und Kosten

scch { }

Nutzen
- Signifikante Reduzierung von Kosten für Fehlersuche und Wartung
- Unterstützt die Standardisierung des Programmdesigns
- Qualitätssteigerung des untersuchten Code

Kosten
- 35% der Use-Case Kosten (65% Förderung), Möglichkeit für das Einbringen von bis zu 15% In-Kind-Leistungen
- Projektkosten (ca.) je Partner 35TEUR (je nach Use-Case).

Aufwand
- Use-Case Definition (vor Projekt)
- Abstimmungsmeetings
- Implementierung



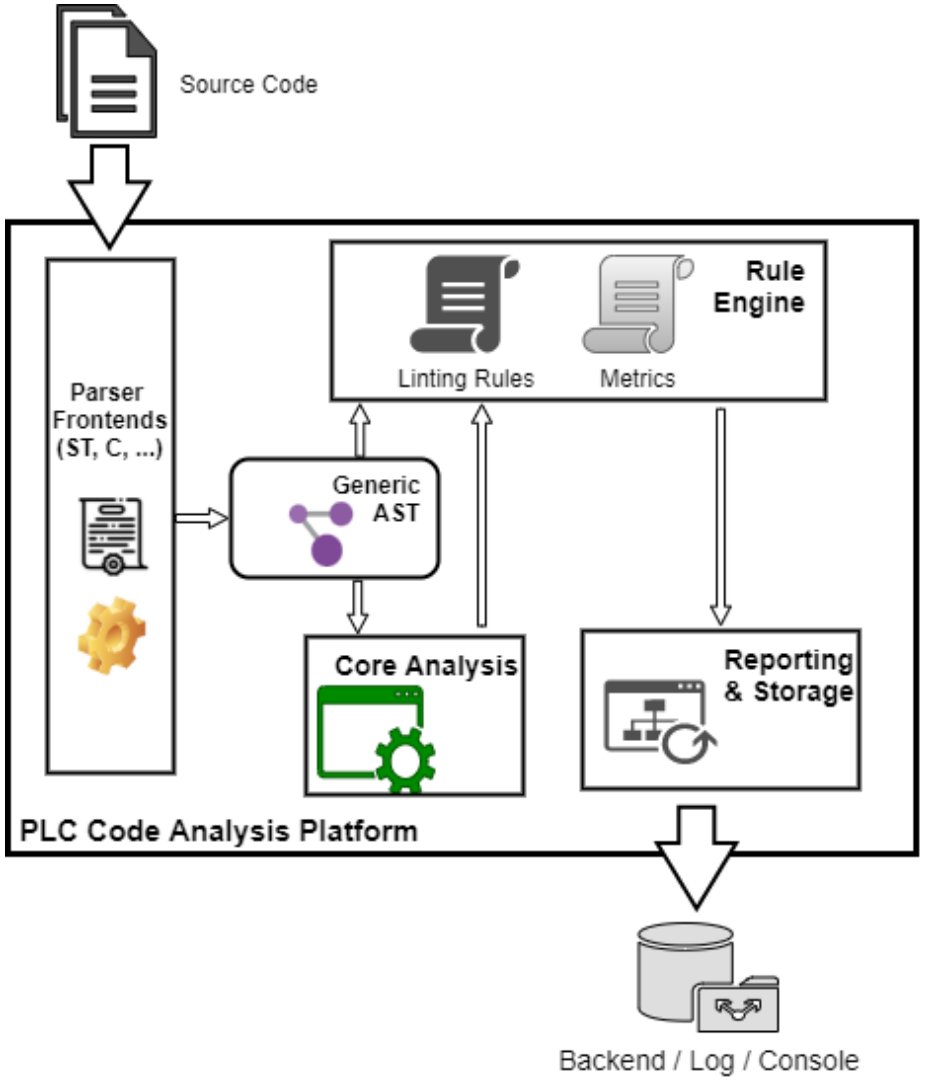
Zusätzlich ohne Mehrkosten!

Erkenntnisse aus den Use-Cases der weiteren Projektpartner

Technik: Analyse von SPS Code

scch { }

- Analyse von SPS Code
  - Unterstützung gängiger Sprachen: IEC-61131 ST/FBD/SFC, ANSI-C, CFC
  - anbieterunabhängig
  - Unterstützung weiterer, auch proprietärer Sprachen möglich
  - basierend auf generischem Modell („Generic AST“)
- Ausführung verschiedener Analysen
  - Grundlegende Analysen („Core Analysis“: Call Graph, Data Flow, Control Flow)
  - Erweiterbare und Konfigurierbare Regel Engine
    - Code Metriken zur Quantifizierung der Qualität
    - Regeln zur Identifikation von Fehlern und anderen Problemen
- liefern Erkenntnisgewinn zu
  - „Code Smells“ und Antipatterns (z.B. unerreichbarer Code)
  - Potentielle Schwachstellen (z.B. aufgrund Security-Richtlinien unerlaubte Funktionen)
  - Potentielle Fehler und Bedrohungen der Stabilität (z.B. Null-Referenzen)
  - Einhaltung von Programmierrichtlinien (z.B. Benennung von Variablen)
  - Erfüllung von Software-Qualitätskriterien
    - Wartbarkeit und Verständlichkeit des Code
    - Identifizierung von Verbesserungspotentialen
- Transparenz durch Reporting
  - Integration in Entwicklungswerkzeuge, aufbereitete Logs
  - Anbindung von Datenbanken zur Speicherung von Analyseergebnissen



Kontaktmöglichkeiten

scch { }

Technische Fragen

Ing. Mag. Berhard Dorninger

bernhard.dorninger@scch.at
+43-50-343 891
www.scch.at

DI Dr. Alois Zoitl

alois.zoitl@jku.at
+43-732-2468 9480
www.jku.at/lit-cyber-physical-systems-lab

Organisatorische Fragen

Ing. DI(FH) Wolfgang Steiner

wolfgang.steiner@biz-up.at
+43-664-8834 7398
www.mechatronik-cluster.at