

CDT bei Bosch Rexroth

(Ein Erfahrungsbericht)

DCC/EDF Harald Kästel-Baumgartner

- Firmenpräsentation
- Ausgangslage
- Zielsetzung
- Know How
- Toolchain
- Projektstruktur
- Positive und negative Erfahrung

Das Unternehmen – Die Produkte

Elektrische
Antriebe
und
Steuerungen



Linear-
technik



Getriebe



Montage-
technik



Hydraulik,
Industrie



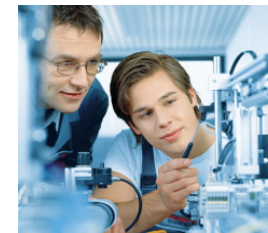
Pneumatik

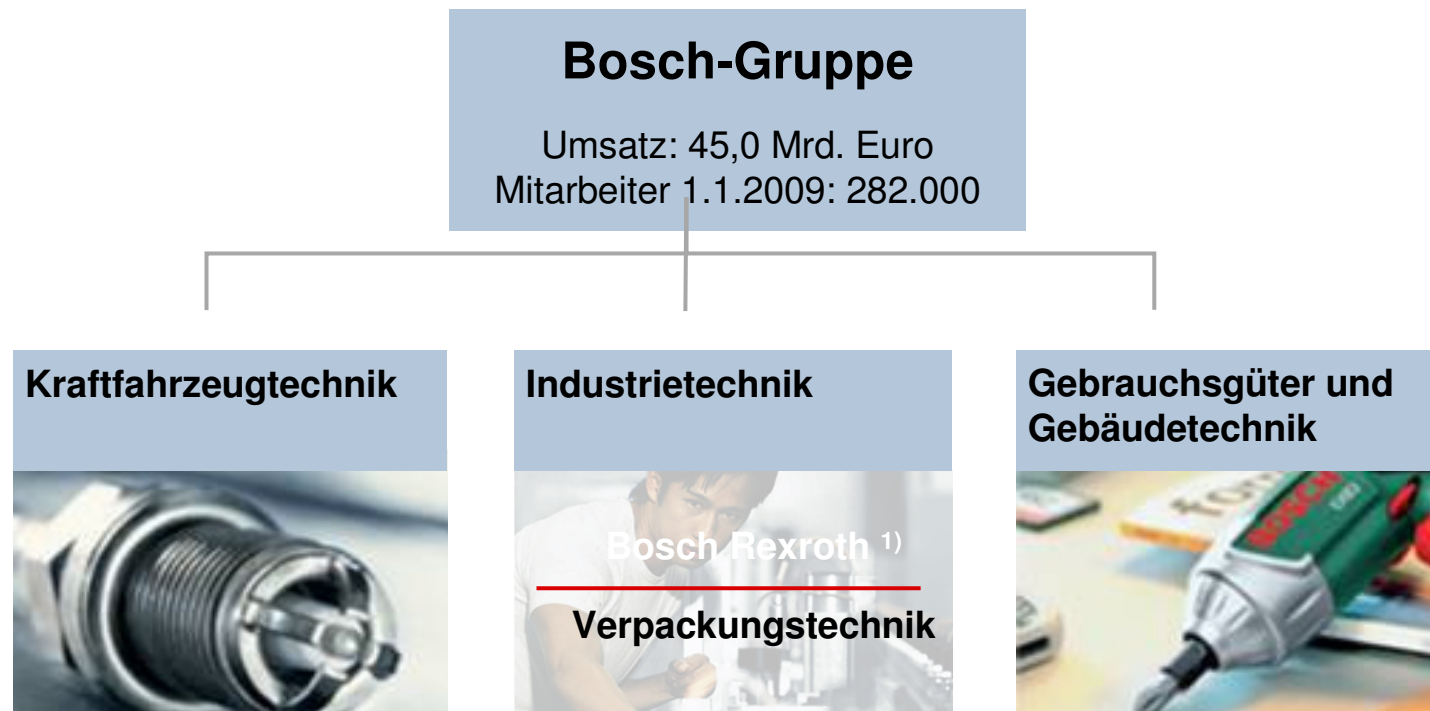


Hydraulik,
Mobil



Service
und
Qualifizierung





¹⁾ Bosch Rexroth AG (100% Bosch)

Elektrische Antriebe und Steuerungen

- Antriebstechnik
- Automationssysteme
- Steuerungstechnik
- Schraub- und Einpress-Systeme
- Widerstandsschweiß-Systeme



- Etablierte Projektstruktur und Vorgehensweise
- Modulorientierte einheitliche Softwarebasis für die unterschiedlichen Derivate der Firmware
- Datenbankbasierte Generierung von Headerdateien
- Präprozessordirektiven für die funktionale Unterscheidung
- Versionsverwaltungssystem
- Skripts für die automatische Erstellung der Derivate
- Entwicklung unter MS Visual Studio ®
- Einsatz diverser Tools zur Qualitätssicherung

- Unterschiedlichste Hardware muss unterstützt werden
- D.h. hohe Flexibilität bei der Entwicklungsumgebung (IDE) gefordert
- Ersatz von MS Visual Studio durch Eclipse CDT
- Unterstützung der etablierten Projektstruktur und Vorgehensweise
- Integration der Datenbanken und Tools in den Buildprozess
- Automatischer Build
- Zentrale interne Softwareverwaltung und Verteilung von Updates

- Erfahrungen mit der Entwicklung eines Plugin im Java Umfeld (JDT)
- „CDT Plug-in Developer Guide“ in der Eclipse Hilfe („Managed Build System Extensibility Document“ / CDT 4.0)
- Diverse Einträge in Bugzilla
- GCC Plugin (org.eclipse.cdt.managedbuilder.gnu.ui) definierte Extensions als Blaupause für die Toolchain
- CDT (org.eclipse.cdt.managedbuilder.core) als Blaupause für Builder und Errorparser
- Sonstiges Material im WWW

- Plugin erstellen auf Basis des CDT Extension Point `org.eclipse.cdt.managedbuilder.core.buildDefinitions`
- Definition von tool, project type, configuration und toolchain
- Hinzufügen von tool options (GUI-Komponenten, Projektdaten)
- Verwendung des Internal Builders (builder, managed build)
- Hinzufügen eines Command Line Generator
- Erstellen eines Error Parser

Praktikable Sonderlösung für das Handling verschiedener Toolchain Versionen (abhängig von der Firmwaregeneration)

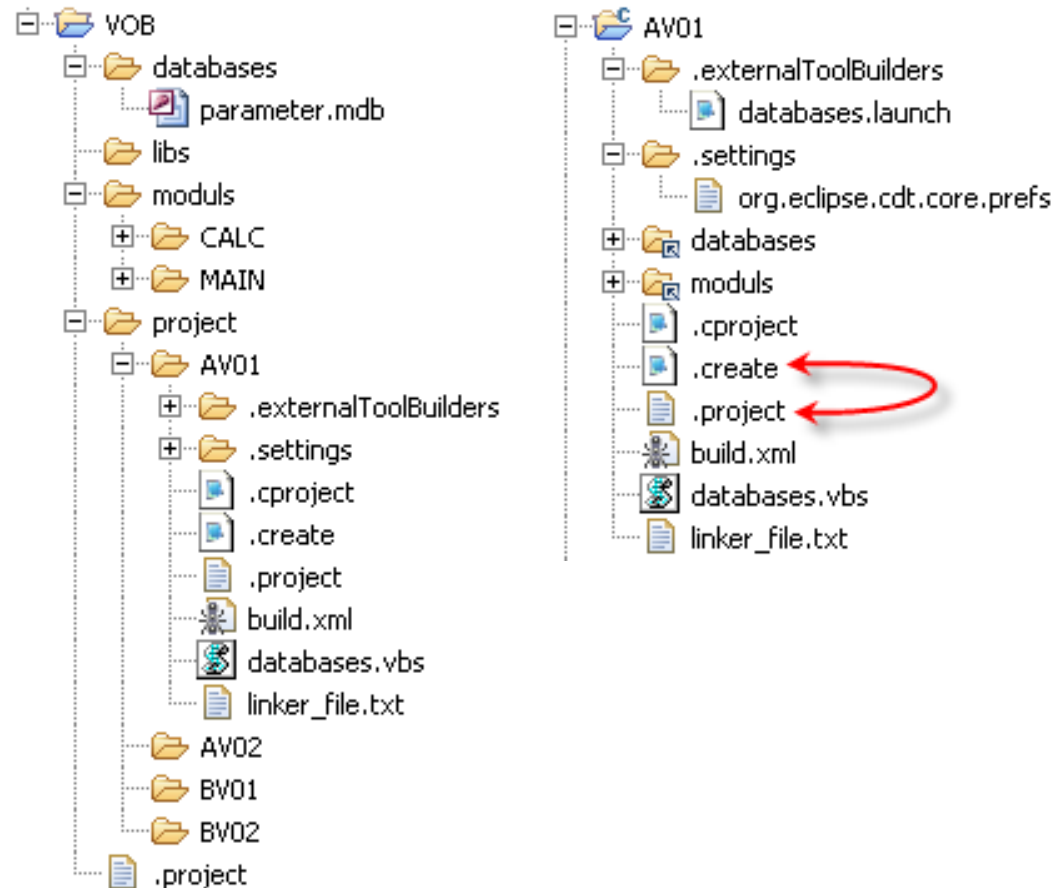
- Vorgegebene plattformabhängige Umgebungsvariablen
- Separate Toolchain Definitionen
- Projektabhängige Verwendung der Umgebungsvariablen

Environment variables to set

Variable	Value	Origin
CWD	D:\Snapsho...	BUILD SYSTEM
PATH	\${SHC_BIN_9}	USER: CONFIG
PC_LINT_HOME	C:\Lint	USER: PREFS
PWD	D:\Snapsho...	BUILD SYSTEM
SHC_7_HOME	C:\HEW7\to...	USER: PREFS
SHC_9_HOME	C:\HEW9\T...	USER: PREFS
SHC_BIN	\${SHC_BIN_9}	USER: CONFIG
SHC_BIN_7	\${SHC_7_H...	USER: PREFS
SHC_BIN_9	\${SHC_9_H...	USER: PREFS
SHC_INC	\${SHC_INC...	USER: CONFIG
SHC_INC_7	\${SHC_7_H...	USER: PREFS
SHC_INC_9	\${SHC_9_H...	USER: PREFS
SHC_LIB	\${SHC_LIB_9}	USER: CONFIG
SHC_LIB_7	\${SHC_7_H...	USER: PREFS
SHC_LIB_9	\${SHC_9_H...	USER: PREFS

Append variables to native environment
 Replace native environment with specified one

- Verlinkte Ressourcen mit relativen Pfaden
- CDT Projektdatei referenziert relative Pfade
- Eclipse Projektdatei .project referenziert absolute Pfade
- Hinzufügen einer neuen Projektdatei .create, um das Projekt plattformunabhängig zu archivieren und wiederherzustellen.

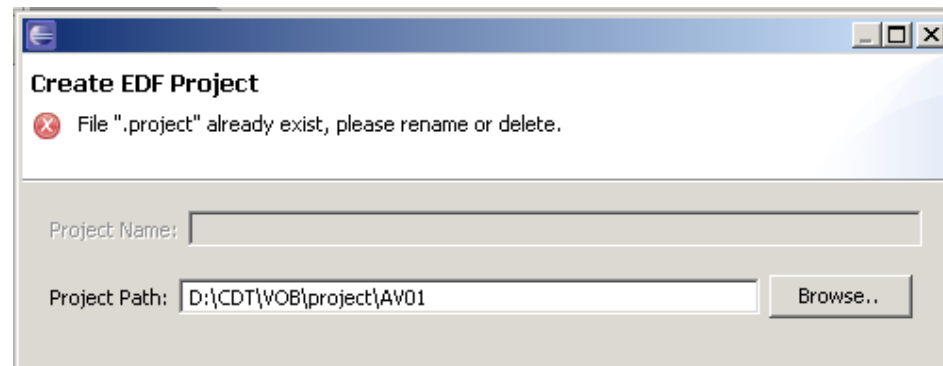


.create als Ersatz für .project

- Plattformunabhängiges Speichern der Projektinformationen
- Basis für das Anlegen des Projekts auf der Entwicklerplattform
- Neuer Wizard

```
<?xml version="1.0" encoding="UTF-8"?>
<project>
  <name>AV01</name>
  <buildCommand>
    <external>/.externalToolBuilders/data
  </buildCommand>
  <buildCommand>
    <name>org.eclipse.cdt.managedbuilder.
  </buildCommand>
  <buildCommand>
    <name>com.boschrexroth.edf.project.pc
  </buildCommand>
  <link>
    <name>databases</name>
    <location>../../databases</location>
  </link>
  <link>
    <name>moduls</name>
    <location>../../moduls</location>
  </link>
</project>
```

```
<extension
  point="org.eclipse.ui.newWizards">
  <wizard
    canFinishEarly="false"
    category="org.eclipse.cdt.ui.newCWizards"
    class="com.boschrexroth.edf.project.CreateProject"
    finalPerspective="org.eclipse.cdt.ui.CPerspective"
    hasPages="true"
    icon="icons/indramat.JPG"
    id="com.boschrexroth.edf.project.wizard"
    name="Create EDF Project"
    project="true">
    <description>
      Erzeugt ein C-Projekt auf der Basis der bereits ...
    </description>
  </wizard>
</extension>
```



```
<project name="sample" default="build" basedir=".">
  <target name="build">
    <cdt.build project="AV01"/>
  </target>
  <macrodef name="cdt.build">
    <attribute name="project"/>
    <sequential>
      <echo>Databases @{project} . . .</echo>
      <exec executable="cscript">
        <arg line="databases.vbs . @{project} full"/>
      </exec>
      <echo>Building @{project} . . .</echo>
      <eclipse.incrementalBuild
        kind      = "full"
        project   = "@{project}"
        builder   = "org.eclipse.cdt.managedbuilder.core.genmakebuilder"/>
      <echo>Finished @{project} . . .</echo>
    </sequential>
  </macrodef>
</project>
```

```
<eclipse path>\eclipse.exe -data <workspace path> -console -noSplash application
org.eclipse.ant.core.antRunner -buildfile build.xml -logfile build.log -vmargs
-Dorg.eclipse.cdt.core.console=org.eclipse.cdt.core.systemConsole
```

- Integration ist einfach möglich durch das Hinzufügen eines zusätzlichen Builders und Error Parsers

```
<extension
  id="com.boschrexroth.nature.edf" name="EDF Project Nature"
  point="org.eclipse.core.resources.natures">
  <runtime> <run class="com.boschrexroth.edf.project.ProjectNature"></run></runtime>
  <builder id="com.boschrexroth.edf.project.pcLintBuilder"></builder>
</extension>
<extension
  id="pcLintBuilder" name="PC Lint Builder"
  point="org.eclipse.core.resources.builders">
  <builder callOnEmptyDelta="false" hasNature="true" isConfigurable="true">
    <run class="com.boschrexroth.edf.project.PCLintBuilder"></run>
  </builder>
</extension>
<extension
  id="com.boschrexroth.edf.project.pcLintErrorParser" name="EDF PC-Lint Error Parser"
  point="org.eclipse.cdt.core.ErrorParser">
  <errorparser class="com.boschrexroth.edf.project.PCLintErrorParser"></errorparser>
</extension>
<extension
  id="pcLintProblem" name="PC Lint Problem" point="org.eclipse.core.resources.markers">
  <super type="org.eclipse.cdt.core.problem"></super>
</extension>
```

```
/**
 * @author harakaes
 * @see org.eclipse.cdt.managedbuilder.internal.core.CommonBuilder
 */
public class PCLintBuilder extends IncrementalProjectBuilder implements
    IResourceVisitor, IResourceDeltaVisitor, IMarkerGenerator {

    private ArrayList<IFile> files;
    private IProject project = null;
    private boolean fullBuild = false;

    protected IProject[] build(int kind, Map args, IProgressMonitor monitor)
        throws CoreException {
        project = getProject();
        IResourceDelta delta = getDelta(project);
        files = new ArrayList<IFile>();
        monitor.beginTask("PC Lint", 100);
        if (kind == FULL_BUILD || delta == null) {
            project.accept(this);
            fullBuild = kind == FULL_BUILD;
        } else {
            delta.accept(this);
            fullBuild = false;
        }
        monitor.worked(10);
        run(new SubProgressMonitor(monitor, 90));
        monitor.done();
        return new IProject[] { project };
    }
}
```

Umgesetzt

- Dekorator für „excluded resources“
- Schlanker ANT-Editor mit reduzierten Java Abhängigkeiten (GUI)

Wunschliste

- Kommando zum Aufruf des Linkers
- Mehr CDT-Kommandos im Navigator (clean und build project)
- Autobuild permanent deaktivieren
- Bessere Verwaltung der Workspaces (Eclipse 3.5, [Bug 162102](#))
- Bessere Indexierung von Headerdateien mit Präprozessordirektiven zur bedingten Kompilierung mit „dynamischen Makrodefinitionen“ (Optionale spezifische Lösung, [Bug 267907](#))
- Besserer Support des Build durch die Eclipse Plattform (org.eclipse.core.resources_3.4.0.v20080604-1400.jar, [Bug 271623](#))

- Sehr gute Erweiterbarkeit um eine Toolchain
- Managed Build System
- Projektdateien sind für Versionsverwaltung geeignet und gut lesbar
- Alle Projektdaten dürfen schreibgeschützt sein
- CodeStyle kann einfach definiert und verteilt werden
- Integration von Dokumentationstools (Doxygen)
- Integration des Versionsverwaltungssystem (ClearCase)
- Professionelle Plattform und Softwareverwaltung
- Vielfalt von Plugins steht zur Verfügung
- Offene Software

- Eclipse ist sehr komplex
- Rudimentäre Verwaltung von Workspaces
- Viel unnötiger Ballast (Automatischer Build, Perspektiven, Views)
- Keine Integration des Betriebssystem-Papierkorbs
- Schlechte Unterstützung eines automatischen Build
- Schwierige Einbindung von Bibliotheken
- Dokumentation und Support ist schlecht

=> Schwieriger Umstieg und schleppende Akzeptanz

Ende

Vielen Dank für Ihre Aufmerksamkeit.