



Richtlinien und deren Prüfung für HANA native Entwicklung (Views / SQLScript)

DSAG Technologietage
Stuttgart, 21. Februar 2018
Jonatan Skopek (MunichRe), Christian Pfaller (CQSE)

- Munich Re Group (MR, ERGO, MEAG, ...) is one of the market leaders in reinsurance business
- Since April 1880

- Key figures (10yr average):
 - Revenue: 44.5 bn €
 - Profit: 2.7 bn €
- Key figures (2016):
 - Employees: about 43.000
 - Equity capital: 31.8 bn €

About Us

Jonatan Skopek  Munich RE 

Business Informatics, Technikum Wien

Engineer - Industrial Engineering

Since 2017: Munich Re – IT Architect

Before: SAP BI Consultant and Developer
(BW, HANA, BODS, BO Frontends)

Dr. Christian Pfaller  CQSE


Master's in Computer Science, TU München

PhD in Software Engineering (Testing)

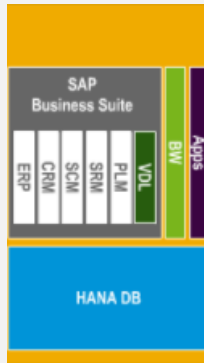
Since 2011: CQSE GmbH

Munich Re: Software Quality Control

Static Code Analysis for ABAP, HANA

Integration in Teamscale 

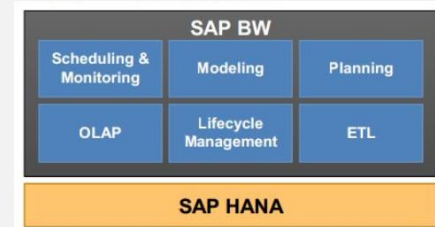
Suite on HANA



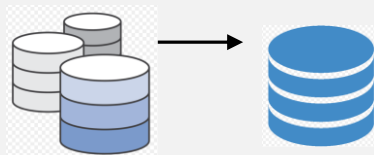
Sparq / Hadoop / Datahub (PoCs)



BW on HANA



Pure data base replacement



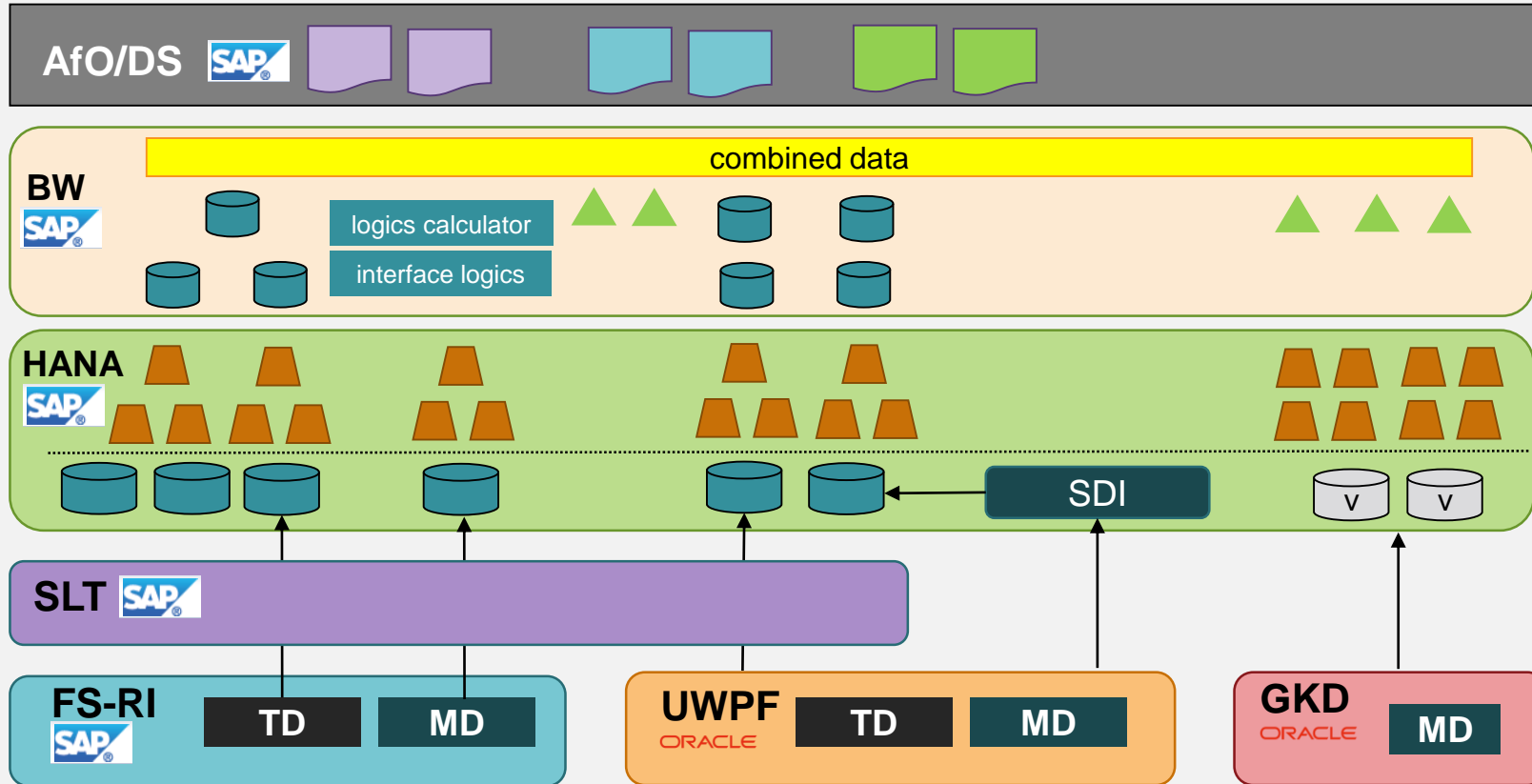
Non-SAP DB



HANA native

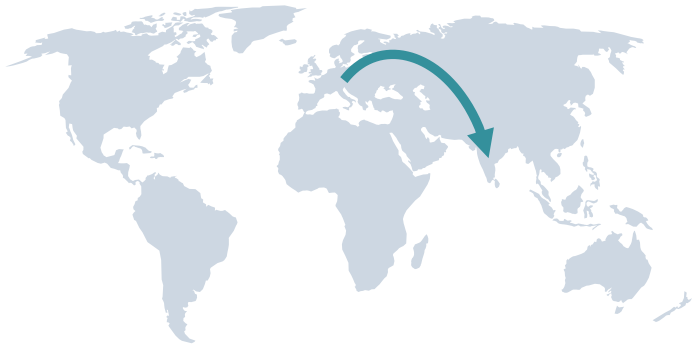


Hybrid Development - BW on HANA / HANA Native



Software Quality – More than „it Works“

Outsourcing / Offshoring



Watch out for Maintainability!

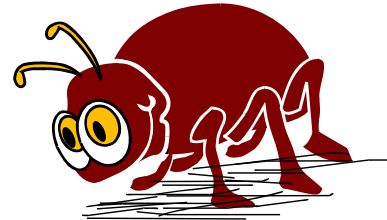


You can Detect a Lot in the Code

Maintainability



Bugs



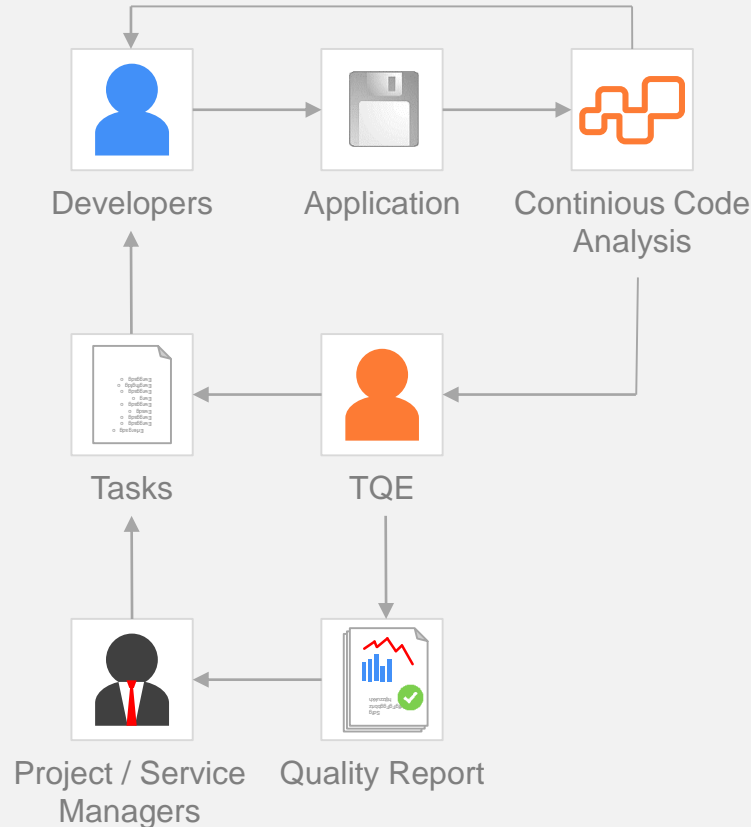
Security



Performance



Technical Quality Engineering



Fast Personal Feedback
by continuous, automatic
code analysis

Detailed Improvements
Suggestions by »TQE«

High Transparency
»Features vs. Quality«
Decisions

SLOC for [blurred] 5.7k -3.3k

Source Objects for [blurred] 162 -87

All Findings Delta for [blurred] 240 7 27

Finding... 0

Code Anomalies for ... 224

Clone Coverage for [blurred]

25.2% +9.90%



Established for ABAP / C# since 2011, aprox. 50 applications

Orphans Delta for [blurred] 0 0 0

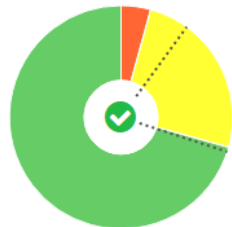
Guideline Violations... 15 2 16

Clone Delta for [blurred] 32 3 6

ject Size fo...



Procedure Length fo...

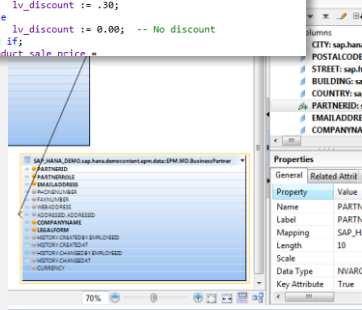


Nesting Depth for [blurred]

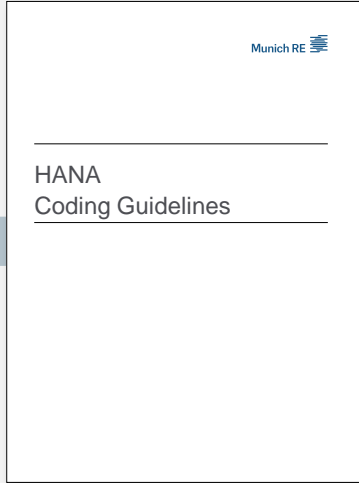



HANA: New Development Technology

```
1 PROCEDURE "SAP_HANA_DEMO"."sap.hana.demontent.epm.Zrscr
2 ( IN productid NVARCHAR(10) )
3 OUT product_sale_price SAP_HANA_DEMO."sap.hana.demontent
4 LANGUAGE SQLSCRIPT
5 SQL SECURITY INVOKER
6 READS SQL DATA AS
7 BEGIN
8 declare lv_category nvarchar(40) := null;
9 declare lv_discount decimal(15,2) := 0;
10 lt_product = select PRODUCTID, CATEGORY, PRICE
11 from "sap.hana.demontent.epm.data::EPM
12 select CATEGORY into lv_category from :lt_product;
13 if :lv_category = 'Notebooks' then
14 lv_discount := .20;
15 elseif :lv_category = 'Handhelds' then
16 lv_discount := .25;
17 elseif :lv_category = 'Flat screens' then
18 lv_discount := .30;
19 elseif :lv_category like '*printers%' then
20 lv_discount := .30;
21 else
22 lv_discount := 0.00; -- No discount
23 end if;
24 product_sale_price :=
```

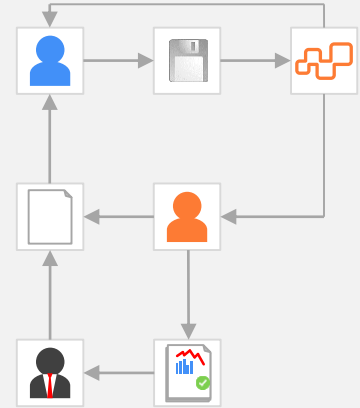


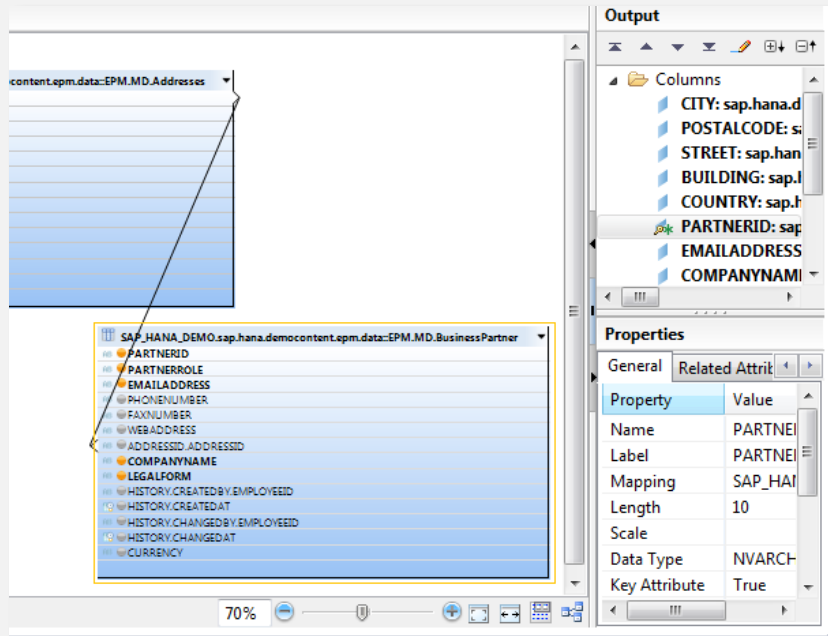
The screenshot shows the SAP HANA Studio interface. On the left, a table structure is displayed with columns: PARTNERID, PARTNERNAME, PARTNERADDRESS, PARTNERCITY, PARTNERSTATE, PARTNERZIP, PARTNERCOUNTRY, PARTNERCOMPANYNAME, PARTNERCOMPANYADDRESS, PARTNERCOMPANYCITY, PARTNERCOMPANYSTATE, PARTNERCOMPANYZIP, PARTNERCOMPANYCOUNTRY, and PARTNERCOMPANYCONTACT. On the right, the 'Properties' pane shows details for the 'PARTNER' table, including Name, Label, Mapping, Length, Scale, Data Type (NVARCHAR), and Key Attribute (True).



Munich RE 

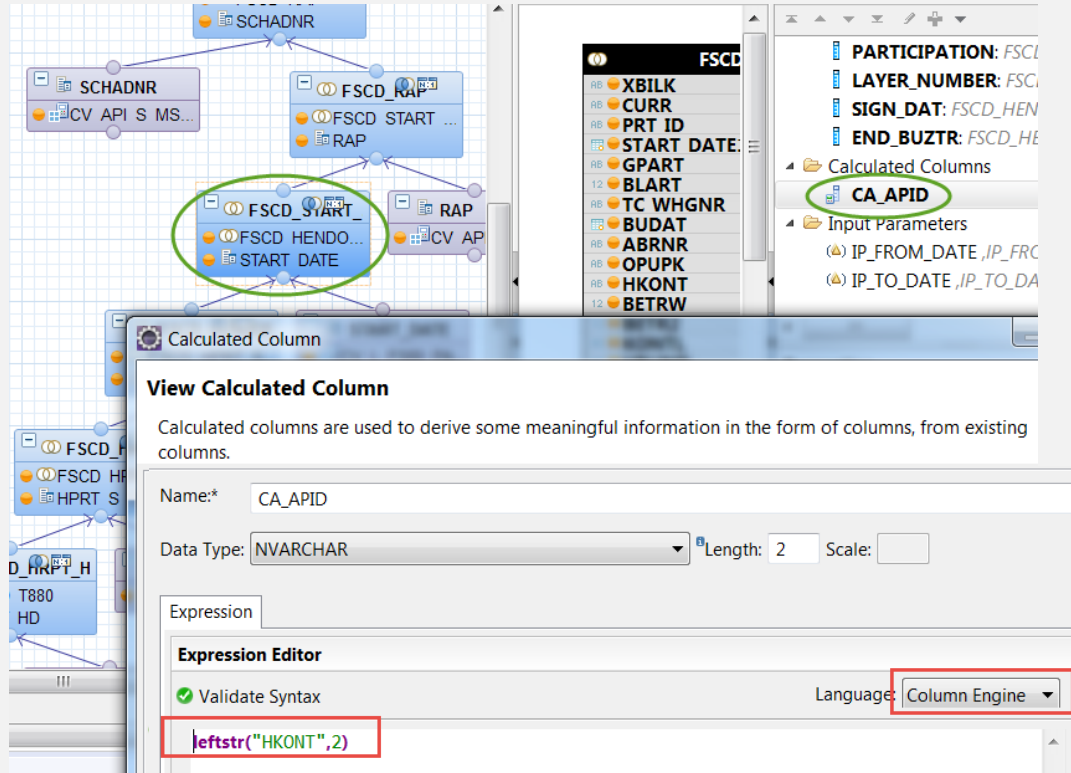
HANA
Coding Guidelines





- Readability
 - Naming of Views
 - Naming of Identifiers
- Robustness
 - No Reserved Words
- Performance
 - No Joins on Calculated Attributes
 - Avoid Column Engine Expressions
- Architecture
 - Ensure Layer Conformance

Performance: Avoid Column Engine



View Calculated Column

Calculated columns are used to derive some meaningful information in the form of columns, from existing columns.

Name*: CA_APID

Data Type: NVARCHAR Length: 2 Scale:

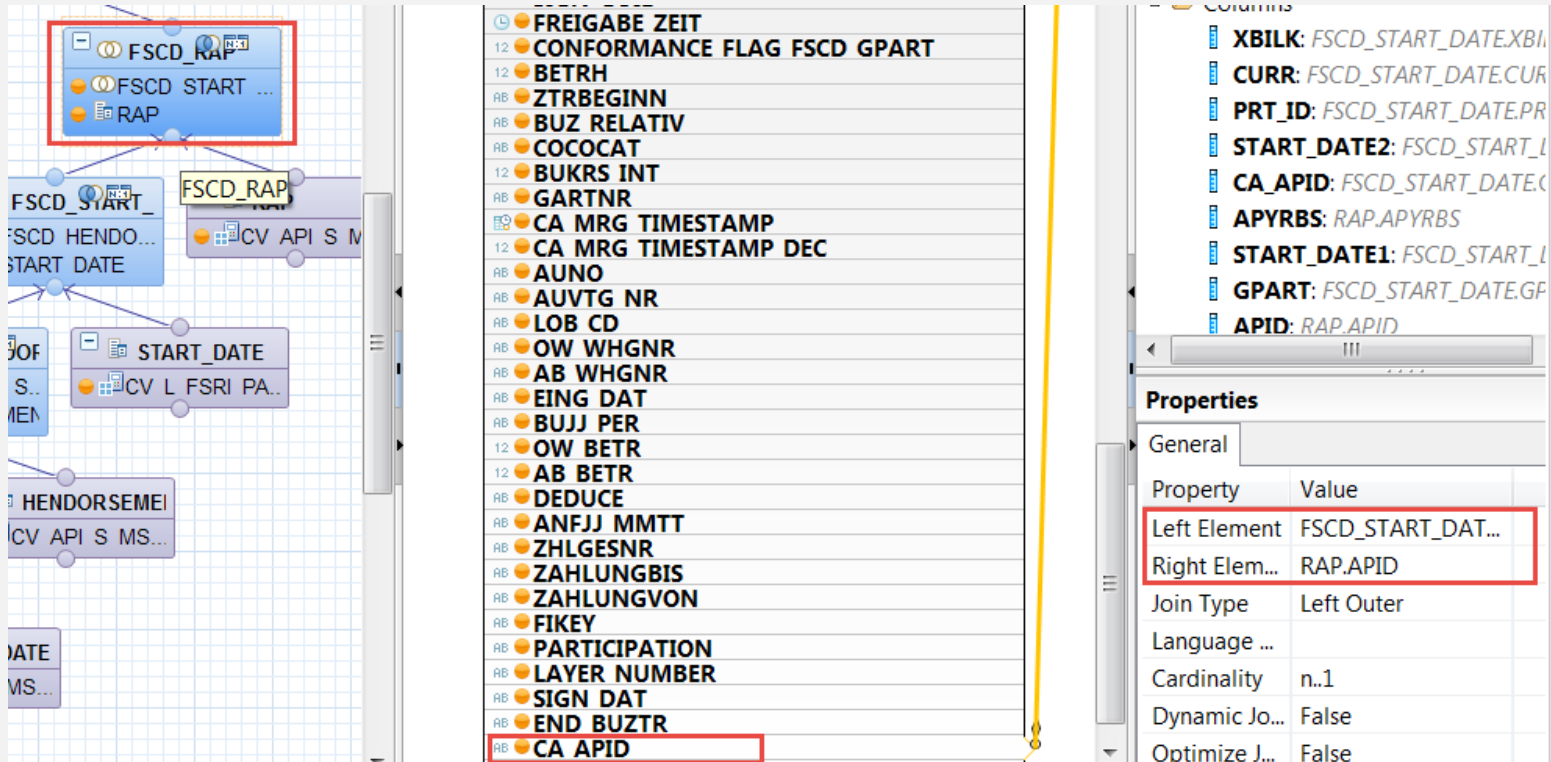
Expression

Expression Editor

Validate Syntax Language: Column Engine

```
leftstr("HKONT",2)
```

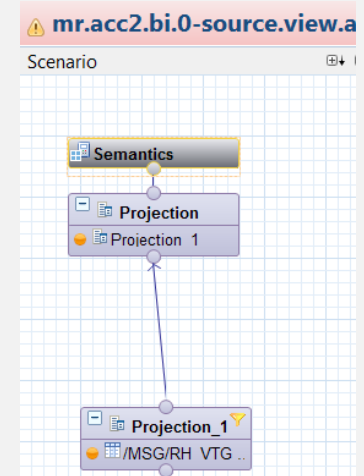
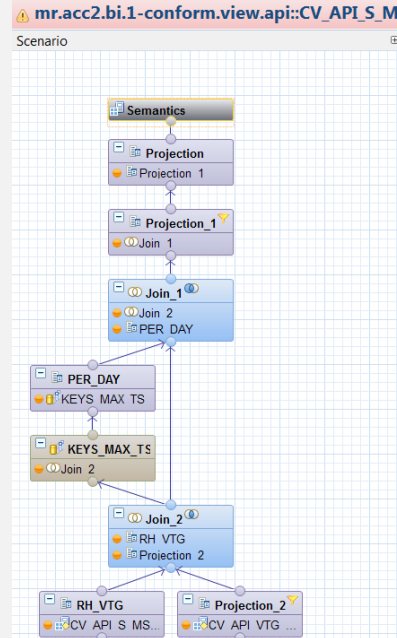
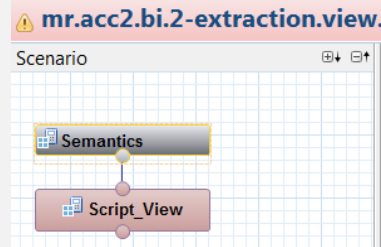
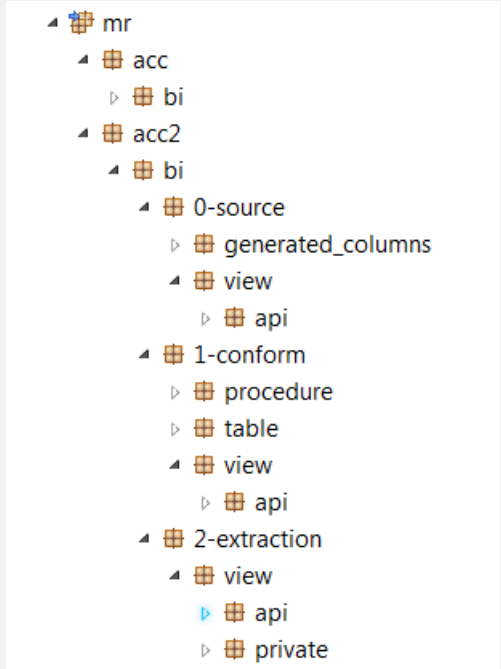
Performance: Avoid Joins on Calculated Attributes



The screenshot displays the SAP ABAP Query Designer interface. On the left, a query plan shows a join node for 'FSCD_RAP' (highlighted with a red box) which is joined to 'FSCD_START...' and 'RAP'. Below it, other nodes like 'FSCD_HENDO...', 'START_DATE', and 'HENDORSEMEI' are visible. The central pane shows a list of columns, with 'CA APID' at the bottom highlighted in red. The right pane shows the 'Columns' list and the 'Properties' section for the join. The 'Properties' section includes a table with the following data:

Property	Value
Left Element	FSCD_START_DAT...
Right Elem...	RAP.APID
Join Type	Left Outer
Language ...	
Cardinality	n..1
Dynamic Jo...	False
Optimize J...	False

Architecture Layers for Views



```
1 PROCEDURE "SAP_HANA_DEMO"."sap_hana_democontent.epm.Proce
2 (IN productid NVARCHAR(10),
3 OUT product_sale_price SAP_HANA_DEMO."sap.hana.democonten
4 LANGUAGE SQLSCRIPT
5 SQL SECURITY INVOKER
6 READS SQL DATA AS
7 BEGIN
8 declare lv_category nvarchar(40) := null;
9 declare lv_discount decimal(15,2) := 0;
10 lt_product = select PRODUCTID, CATEGORY, PRICE
11 from "sap.hana.democontent.epm.data::EPM
12 select CATEGORY into lv_category from :lt_product;
13 if :lv_category = 'Notebooks' then
14 lv_discount := .20;
15 elseif :lv_category = 'Handhelds' then
16 lv_discount := .25;
17 elseif :lv_category = 'Flat screens' then
18 lv_discount := .30;
19 elseif :lv_category like '%printers%' then
20 lv_discount := .30;
21 else
22 lv_discount := 0.00; -- No discount
23 end if;
24 product_sale_price =
25 select PRODUCTID, CATEGORY, PRICE,
26 PRICE - cast((PRICE * :lv_discount) as decim
27 from :lt_product;
28 END;
```

- Readability
 - Avoid Nested Subqueries
 - Nested Control Blocks
 - Long Procedures
- Robustness
 - Missing LANGUAGE definition
 - Missing Quotes at Identifiers
- Performance
 - Detect Imperative Control Statements
- Duplicated Code

Nested Subqueries in SQL Statements

```
SELECT AC."MANDT", AC."PRT_SEQ_GUID", AC."AEDAT" || AC."AEZET" || '0000000' AS "TS", AC."EXT_REF_PR  
'I' AS "H_KZ"  
FROM "_SYS_BIC"."sqlscript.checks.test/CV_API_S_MSG_EC_HPRT01" AC  
INNER JOIN  
(  
    SELECT "MANDT", "PRT_SEQ_GUID"  
    FROM(  
        SELECT A."MANDT", A."PRT_SEQ_GUID", B."EX_FLAG" FROM  
        (  
            SELECT C."MANDT", C."PRT_SEQ_GUID"  
            FROM  
                (SELECT MANDT, PRT_SEQ_GUID FROM "_SYS_BIC"."sqlscript.checks.test/CV_API_S_MS  
                INNER JOIN  
                    (SELECT MANDT, PRT_SEQ_GUID FROM "_SYS_BIC"."sqlscript.checks.test/CV_API_S_MS  
                ON C.MANDT = D.MANDT  
                AND C.PRT_SEQ_GUID = D.PRT_SEQ_GUID  
            ) A  
        LEFT OUTER JOIN  
        (  
            SELECT MANDT, PRT_SEQ_GUID, 1 AS EX_FLAG  
            FROM  
                (  
                    SELECT "MANDT", "PRT_SEQ_GUID" FROM #"GRANULAR_DATA" GROUP BY MANDT, PRT_SEQ_GUID
```



Duplicated Code

```
299 /*-----
300 -- CONTAINED_IN_REPORT = 0
301 1. EXPECTED_FIELD - NULL
302 Applicable for Pricing team only
303 -----*/
304
305
306 IF cur_row."CONTAINED_IN_REPORT" = 0 and cur_row."EXPECTED_FIELD" IS NULL THEN
307
308 CREATE LOCAL TEMPORARY TABLE #tloc1 (tcount1 INTEGER);
309 v_select_count := :v_select_count || 'INSERT INTO #tloc1 SELECT COUNT(*) AS tcount1
310
311 EXEC v_select_count;
312
313 SELECT DISTINCT(tcount1) into a_count from #tloc1;
314 DROP table #tloc1;
315
316 IF :a_count != 0 THEN
317 INSERT INTO "MR_COM_TESTFWK"."mr.com.testfwk.0-source.table::TEST_RESULTS"("TEST_AS
318 VALUES (cur_row."DESCRIPTION", 'FAIL',CURRENT_TIMESTAMP,cur_row."TS_ID",cur_ro
319 ELSE
320 INSERT INTO "MR_COM_TESTFWK"."mr.com.testfwk.0-source.table::TEST_RESULTS"("TEST_AS
321 VALUES (cur_row."DESCRIPTION", 'PASS',CURRENT_TIMESTAMP,cur_row."TS_ID",cur_ro
322 END IF;
323
324 END IF;
325
326 -----
327 -- Procedure Execution Log --
328 -- For Technical analysis purposes --
```

```
191 -----
192
193 --2. EXPECTED_FIELD - NULL
194 -----
195
196 IF cur_row."CONTAINED_IN_REPORT" = 1 AND
197 cur_row."EXPECTED_FIELD" IS NULL THEN
198
199 -- Check the validity of test case; though EXPECTED_FIELD is NULL
200 CREATE LOCAL TEMPORARY TABLE #tloc (tcount INTEGER);
201 v_select_count := :v_select_count || 'INSERT INTO #tloc SELECT COUNT(*) AS tcount FF
202
203 EXEC v_select_count;
204
205 SELECT DISTINCT(tcount)
206 INTO a_count
207 FROM #tloc;
208
209 DROP table #tloc;
210
211 -- PASS if result row count = 1; otherwise log it as FAILED
212 IF :a_count != 1 THEN
213 INSERT INTO "MR_COM_TESTFWK"."mr.com.testfwk.0-source.table::TEST_RESULTS"("TEST_/
214 VALUES (cur_row."DESCRIPTION", 'FAIL',CURRENT_TIMESTAMP,cur_row."TS_ID",cur_r
215 ELSE
216 INSERT INTO "MR_COM_TESTFWK"."mr.com.testfwk.0-source.table::TEST_RESULTS"("TEST_/
217 VALUES (cur_row."DESCRIPTION", 'PASS',CURRENT_TIMESTAMP,cur_row."TS_ID",cur_r
218 END IF;
219 END IF;
220
```

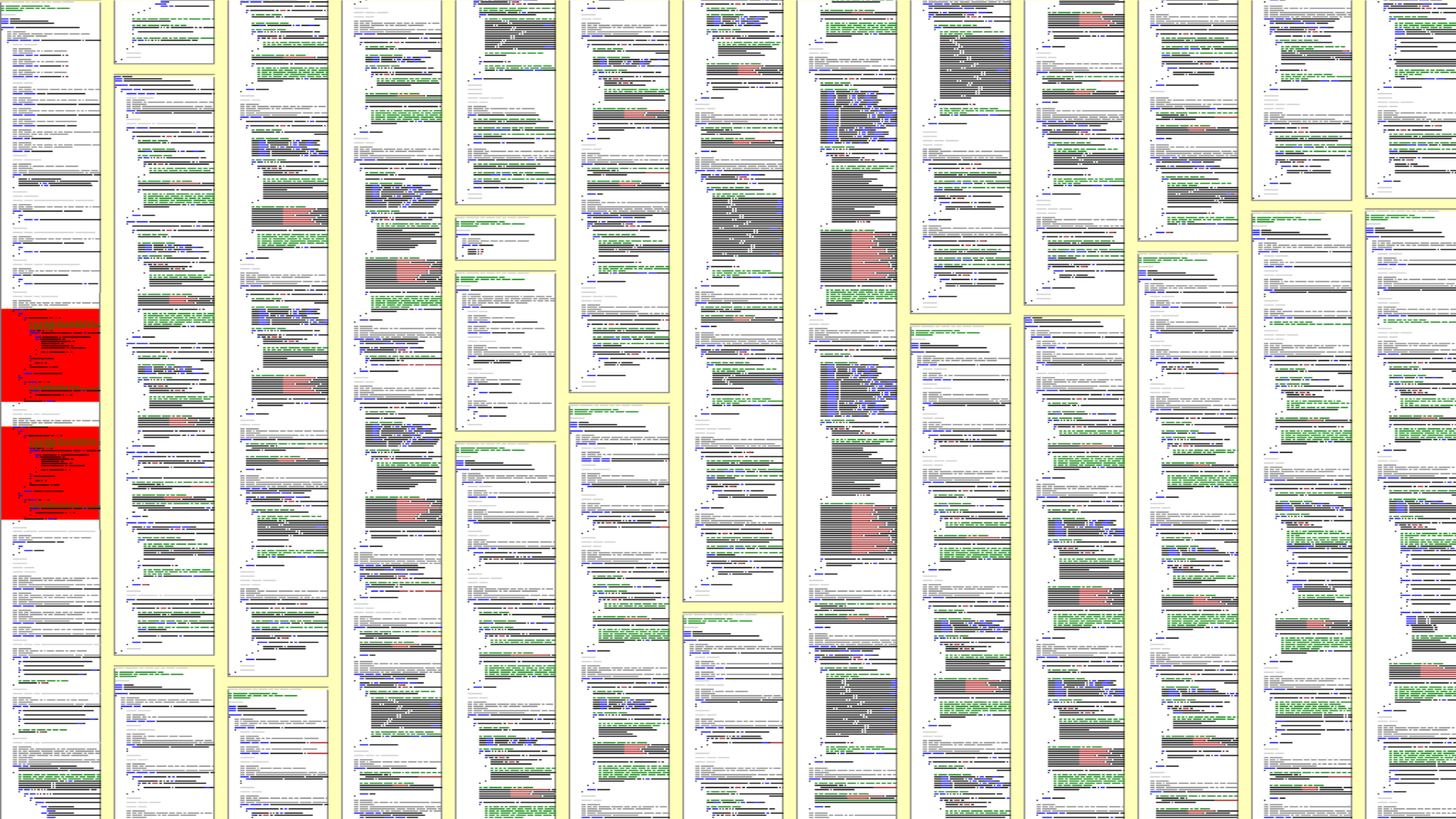
Inconsistent Changes: Risk of Bugs

```
// Utilities for arrays of elements
public String showElements(ModelElement[] elements, String nomsg) {
    boolean found = false;
    StringBuffer res = new StringBuffer();
    if (elements != null) {
        Index.getInstance().setCurrentRenderer(
            FlatReferenceRenderer.getInstance());
        for (int i = 0; i < elements.length; i++) {
            ModelElement el = elements[i];
            res.append(showElementLink(el)).append(HTML.LINE_BREAK);
            found = true;
        }
        Index.getInstance().resetCurrentRenderer();
    }
    if (!found && nomsg != null && nomsg.length() > 0) {
        res.append(HTML.italics(nomsg));
    }
    return res.toString();
}
```

```
// Utilities for arrays of elements
public String showElements(ModelElement[] elements, String nomsg) {
    boolean found = false;
    StringBuffer res = new StringBuffer();
    if (elements != null) {
        Index.getInstance().setCurrentRenderer(
            FlatReferenceRenderer.getInstance());
        for (int i = 0; i < elements.length; i++) {
            ModelElement el = elements[i];
            res.append(showElementLink(el)).append(HTML.LINE_BREAK);
            found = true;
        }
        Index.getInstance().resetCurrentRenderer();
    }
    if (!found && nomsg.length() > 0) {
        res.append(HTML.italics(nomsg));
    }
    return res.toString();
}
```

```
// Utilities for arrays of elements
public String showElements(ModelElement[] elements, String nomsg) {
    boolean found = false;
    StringBuffer res = new StringBuffer();
    if (elements != null) {
        Index.getInstance().setCurrentRenderer(
            FlatReferenceRenderer.getInstance());
        for (int i = 0; i < elements.length; i++) {
            ModelElement el = elements[i];
            res.append(showElementLink(el)).append(HTML.LINE_BREAK);
            found = true;
        }
        Index.getInstance().resetCurrentRenderer();
    }
    if (!found && nomsg != null && nomsg.length() > 0) {
        res.append(HTML.italics(nomsg));
    }
    return res.toString();
}
```







Scientific Study

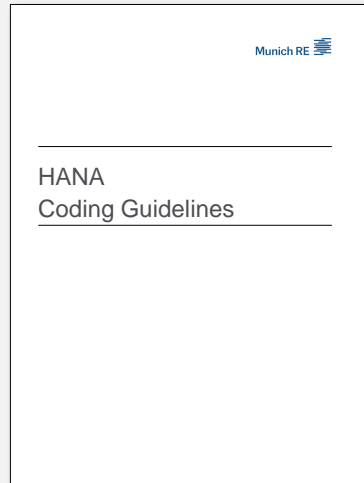
Munich RE   

- More than 100 bugs in production



- 52% of all unintended inconsistencies

Juergens, Deissenboeck et al: *Do Code Clones Matter?* ICSE 2009



- ABAP / C#




- Extend for
 - SQLScript
 - HANA Views (XML)
- Access HANA Development Objects?

How to Get the Code?

HANA
Code Repository

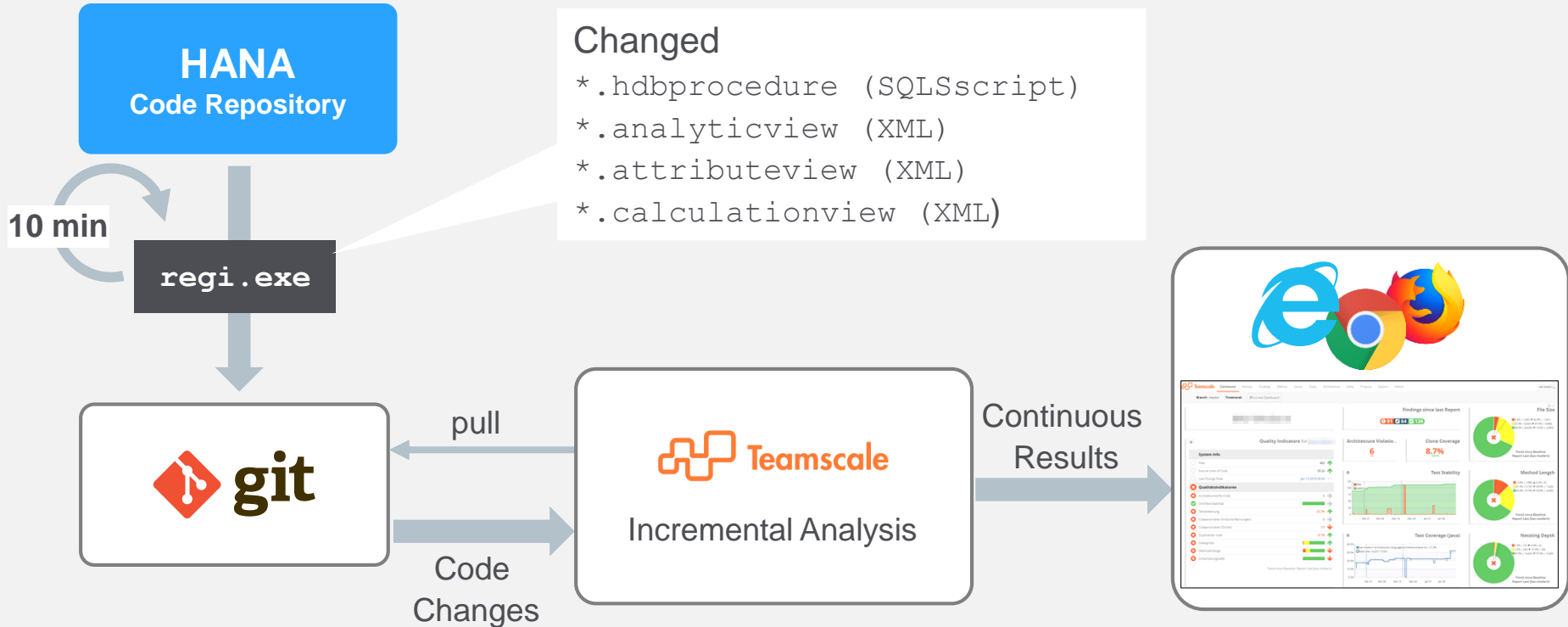



Incremental Analysis

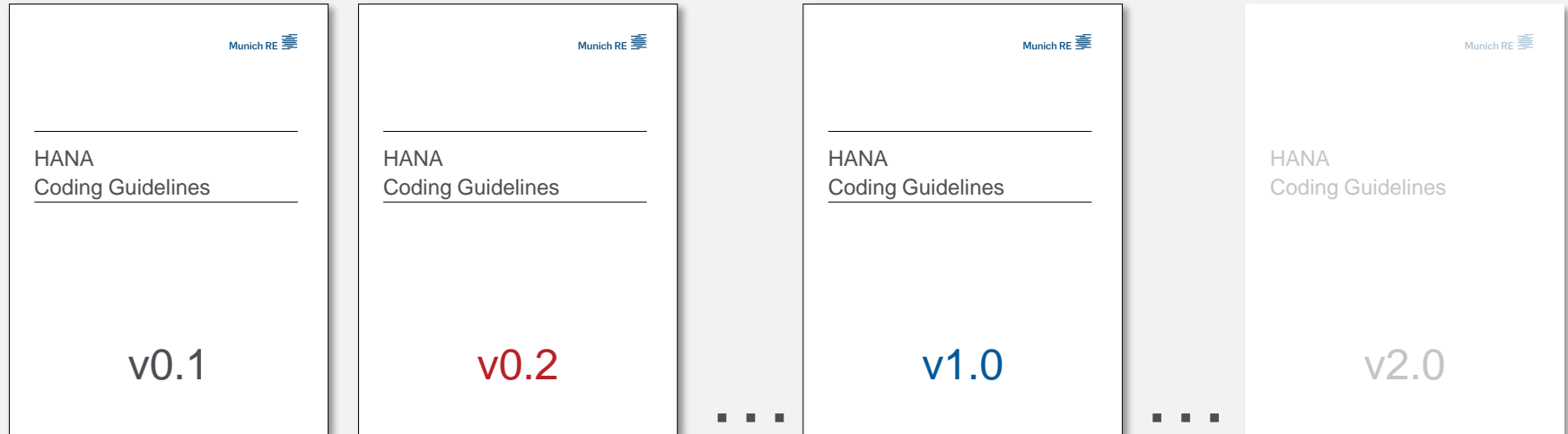
Continuous
Results



How to Get the Code?

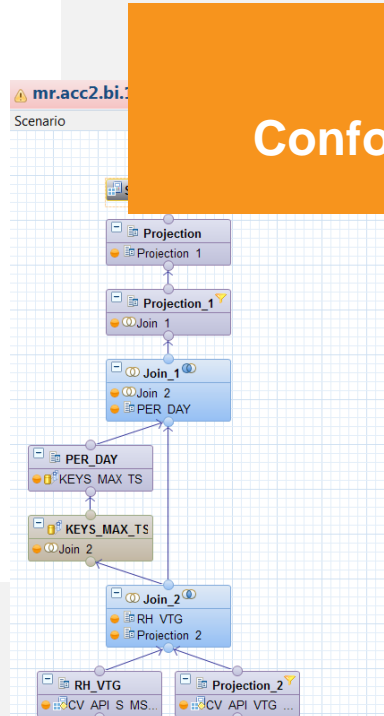


Guidelines Evolve ...

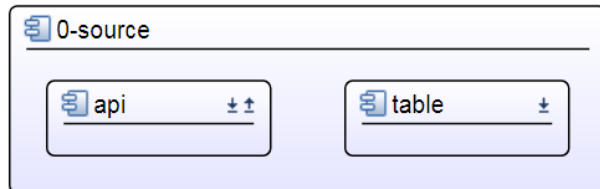
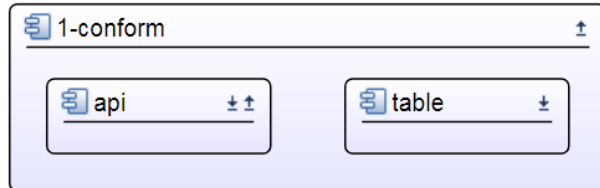
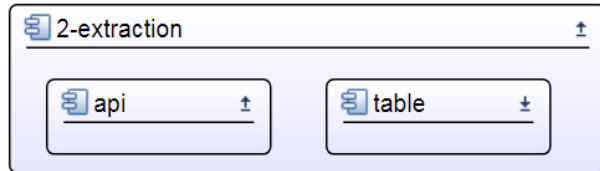


```
14 @ACheck(name = "Avoid cursors", description = "Cursors imply row-at-a-time processing. As a "
15     + "consequence, opportunities for optimizations by the SQL engine are missed.",
16     groupName = ISqlScriptCheckGroups.UNWANTED_SQL_SCRIPT_STATEMENTS,
17     languages = { ELanguage.SQLSCRIPT }, parameters = { ECheckParameter.ABSTRACT_SYNTAX_TREE })
18 public class AvoidCursorCheck extends EntityFindingCheckBase {
19
20     @Override
21     protected String getXPathSelectionString() {
22         return "//ATTRIBUTE[subtype(' + SubTypeNames.CURSOR + ')]";
23     }
24
25     @Override
26     protected String getFindingMessage(ShallowEntity entity) {
27         String message = "Usage of cursor";
28         List<IToken> tokens = entity.ownStartTokens();
29         if (tokens.size() > 2) {
30             String identifier = tokens.get(2).getText();
31             message += " '" + identifier + "'";
32         }
33         return message;
34     }
35 }
```

- mr
 - acc
 - bi
 - acc2
 - bi
 - 0-source
 - generated_columns
 - view
 - api
 - 1-conform
 - procedure
 - table
 - view
 - api
 - 2-extraction
 - view
 - api
 - private



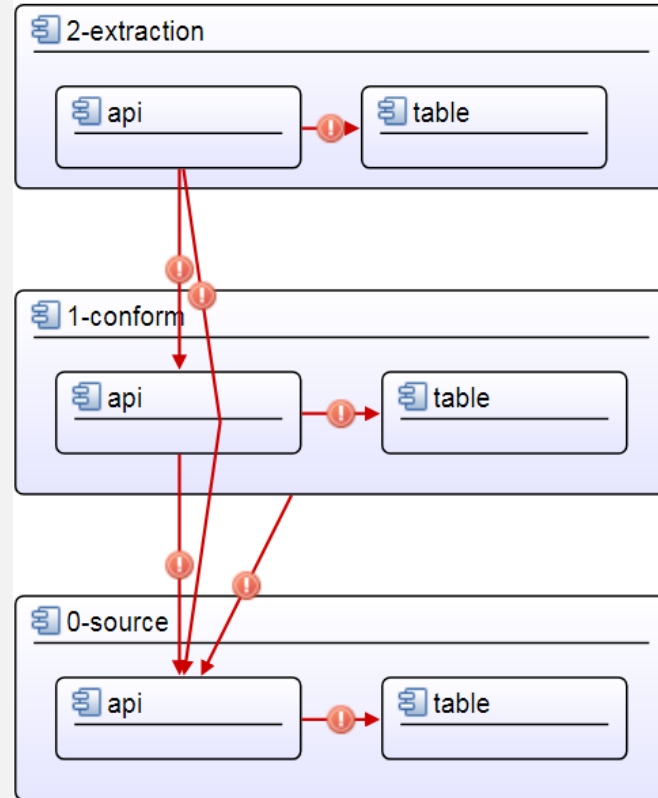
**Views Inside:
Conform with Layer Architecture?**



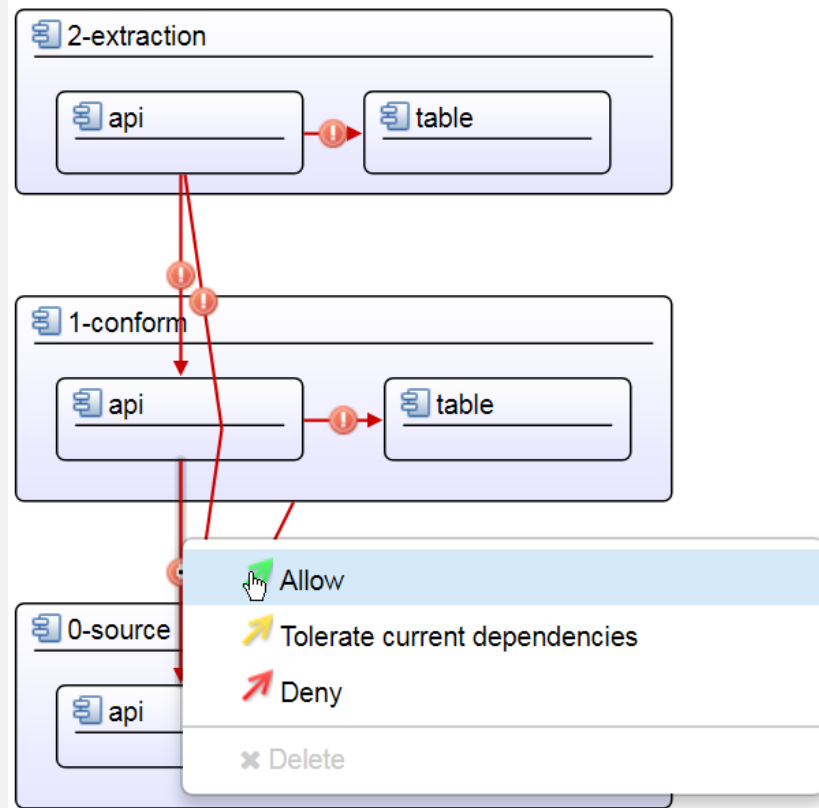
Mappings: ⊕

- include mr\acc\bi\1\conform\.*
- include mr\acc2\bi\1\conform\.*
- include acc\bi\1\conform\.*
- include acc2\bi\1\conform\.*
- exclude .*\(view\api|table)\.*

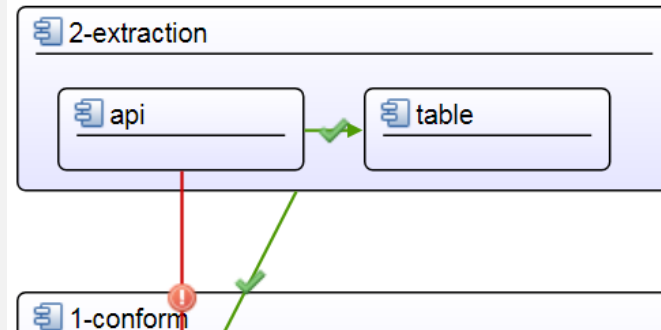
Modelling Intended Architecture – Dependencies



Modelling Intended Architecture – Dependencies

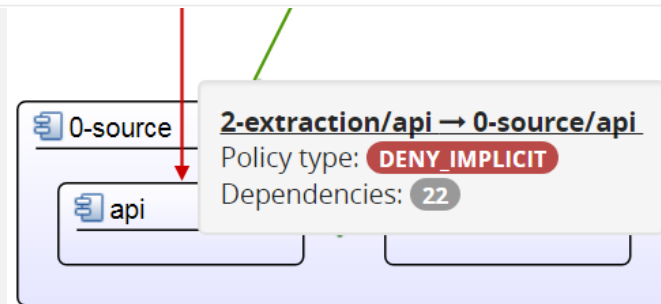


Showing Violations



Dependencies: 2

Source	Target
acc.bi.2-extraction.view.api.text.calculationviews.CV_API_ACRMUWID_TEXT	mr.acc.bi.0-source.view.api.text.calculationviews.CV_API_S_MSG_HPOLICY_HD_TEXT
acc.bi.2-extraction.view.api.text.calculationviews.CV_API_MSG_HPOLICY_HD_TEXT	mr.acc.bi.0-source.view.api.text.calculationviews.CV_API_S_MSG_HPOLICY_HD_TEXT



Type depends on 'mr.acc2.bi.0-source.view.api.calculationviews.CV_API_S_MSG_RVTG'. This violates the architecture specification in 'dsag_demo.architecture'

[Architecture](#) / [Architecture Conformance](#)

A violation of a dependency policy in the architecture definition.

Blacklisted

Yes

No

Details

introduced Feb 15 2018 16:47

no diff (introduced by changes in other file)

[Diff to...](#)

Associated Tasks

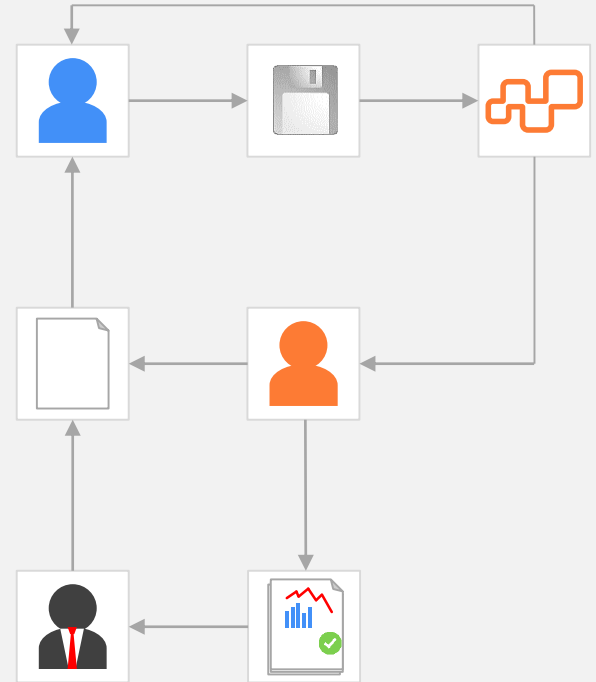
None

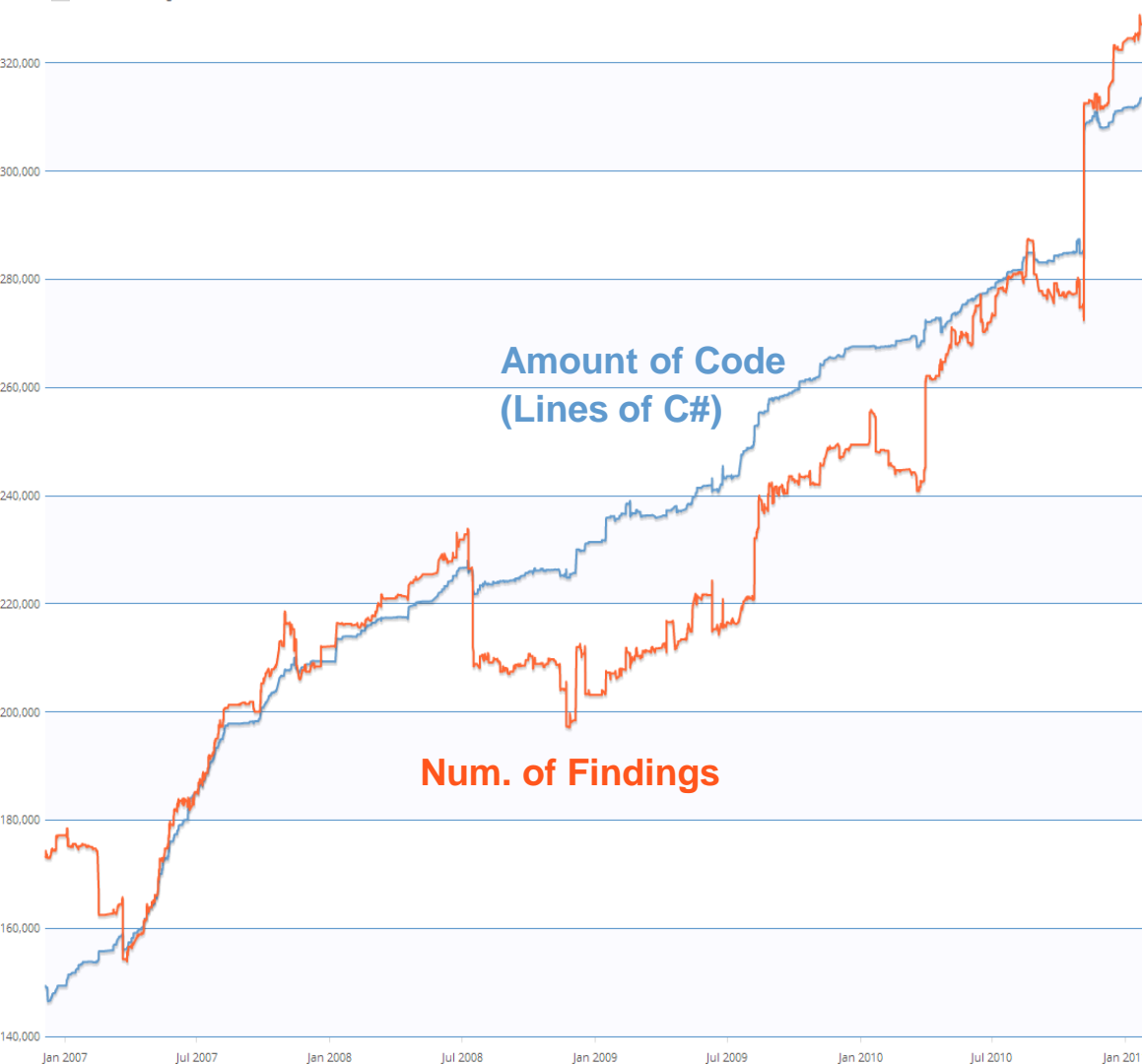
[Add to Task](#)

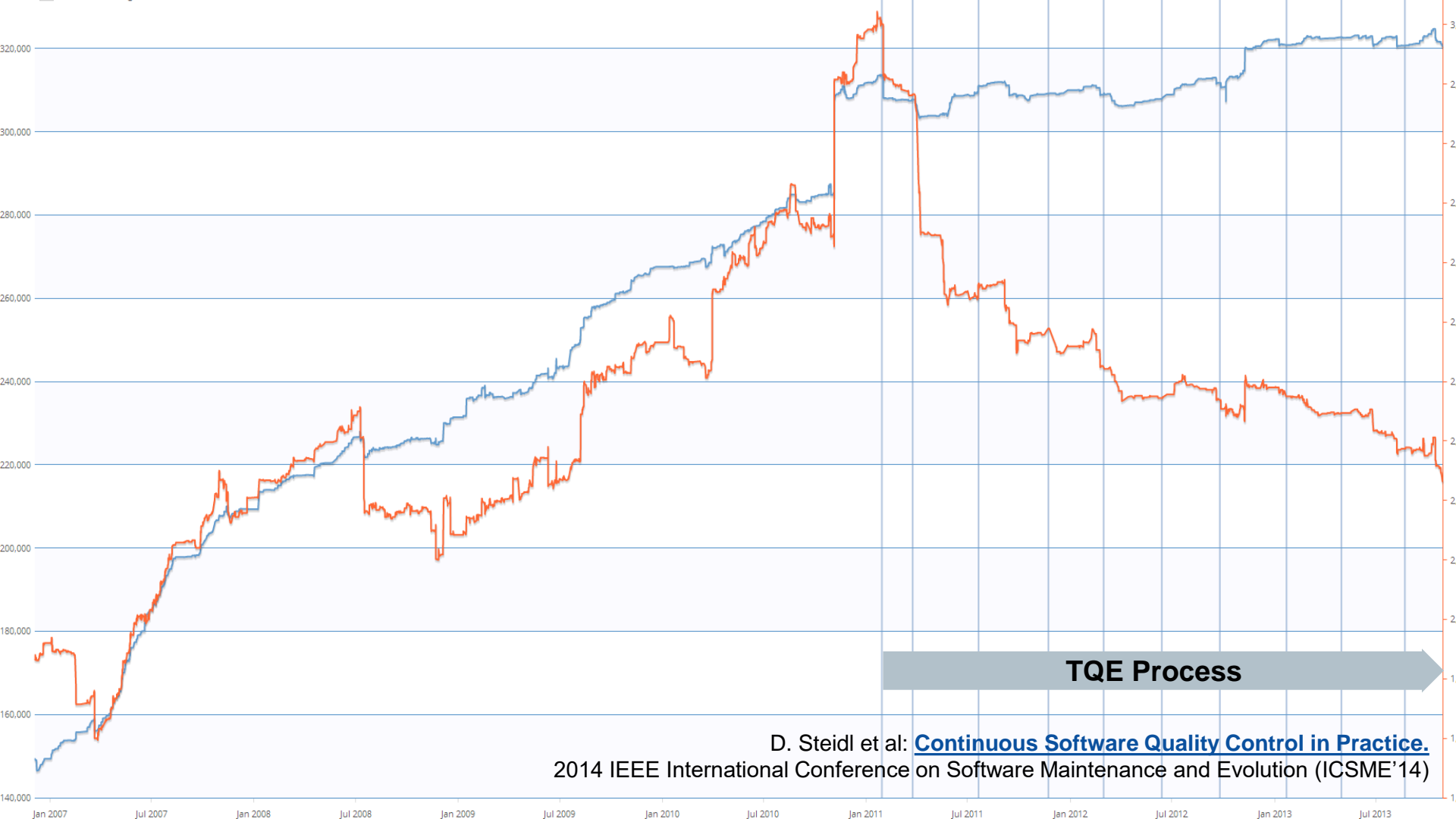
in [acc2/bi/2-extraction/view/api/CV_L_CSPI_BASE.calculationview](#) (Lines: 10)

```
7      <resourceUri>  
8      <DataSource id="CV_API_S_MSG_RVTG" type="CALCULATION_VIEW">  
9          <viewAttributes allViewAttributes="true"/>  
10         <resourceUri>/mr.acc2.bi.0-source.view.api/calculationviews/CV_API_S_MSG_RVTG</resourceUri>  
11     </DataSource>  
12     <DataSource id="CV_API_S_MSG_RVTGANT" type="CALCULATION_VIEW">  
13         <viewAttributes allViewAttributes="true"/>  
14         <resourceUri>/mr.acc2.bi.1-conform.view.api/calculationviews/CV_API_S_MSG_RVTGANT</resourceUri>  
15     </DataSource>  
16     <DataSource id="CV_API_S_MSG_RVTGANTKO" type="CALCULATION_VIEW">  
17         <viewAttributes allViewAttributes="true"/>  
18         <resourceUri>/mr.acc2.bi.1-conform.view.api/calculationviews/CV_API_S_MSG_RVTGANTKO</resourceUri>
```

TQE Process, Static Code Analysis: Observed Effect







TQE Process

D. Steidl et al: [Continuous Software Quality Control in Practice](#).
2014 IEEE International Conference on Software Maintenance and Evolution (ICSME'14)

- Important to define design guidelines
- Continuous update due to technology changes and growing experience
- Two major purposes for code quality checks:
 - Self service tool for developers to fulfill design guidelines already during initial development
 - Code quality reports of already implemented code
- Checks should be part of development process (quality gates, etc.)
- Automated checks beneficial especially in large development projects

Questions welcome!

Jonatan Skopek, JSkopek@munichre.com, +49 89 3891-3569

Dr. Christian Pfaller, pfaller@cgse.eu, +49 176 10090694

Blog-Post on SQLScript Checks: <https://www.cgse.eu/en/blog/sqlscript-analysis/>