

# Analytische Software Qualitätssicherung

**Eine Ringvorlesung von  
Harry M. Sneed  
SoRing Kft., Budapest  
für die TU Dresden  
21. November 2016**

# Ansätze zur Software-Qualitätssicherung

- **Psychologische Ansätze**
- **Konstruktive Ansätze**
- **Analytische Ansätze**
- **Empirische Ansätze**
- **Sanierungsansätze**

# Psychologische Ansätze der Software-Qualitätssicherung

- \* **Mitarbeitermotivation**
- \* **Preußische Beamtenethik**
- \* **Gewinnbeteiligung**
- \* **Earned Values**
- \* **Mitarbeiter Ratings**
- \* **Wartungsdienst**
- \* **Quality Circles**
- \* **Statistische Projektüberwachung**

# Konstruktive Ansätze der Software-Qualitätssicherung

- \* **Schulung der Mitarbeiter**
- \* **Fundierte Entwicklungsmethodik**
- \* **Formale Fachkonzepte**
- \* **Normierte Entwurfsdokumentation**
- \* **Höhere Programmiersprachen**
- \* **Wiederverwendbare Programmrahmen**
- \* **Integriertes Repository**
- \* **CASE-Werkzeuge**
- \* **Objektorientierte Architekturen**

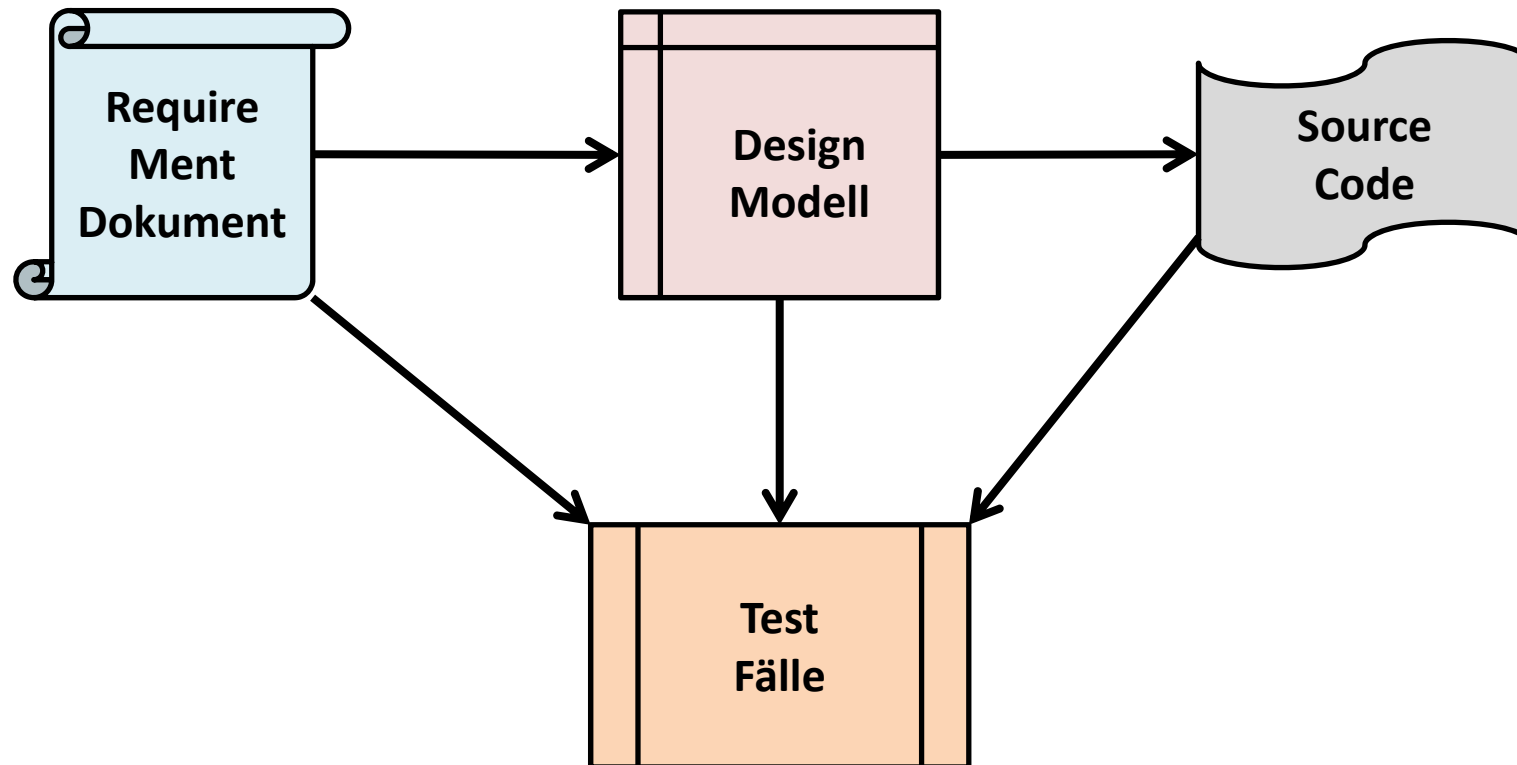
# Analytische Ansätze der Software-Qualitätssicherung

- \* **Verifikationstechniken**
- \* **Beweisführung**
- \* **Konformitätsprüfung**
- \* **Reviews**
- \* **Walk Throughs**
- \* **Inspektionen**
- \* **Statische Analyse**
- \* **Dynamische Analyse**
- \* **Animation**
- \* **Abnahmeverfahren**

# Empirische Ansätze der Software-Qualitätssicherung

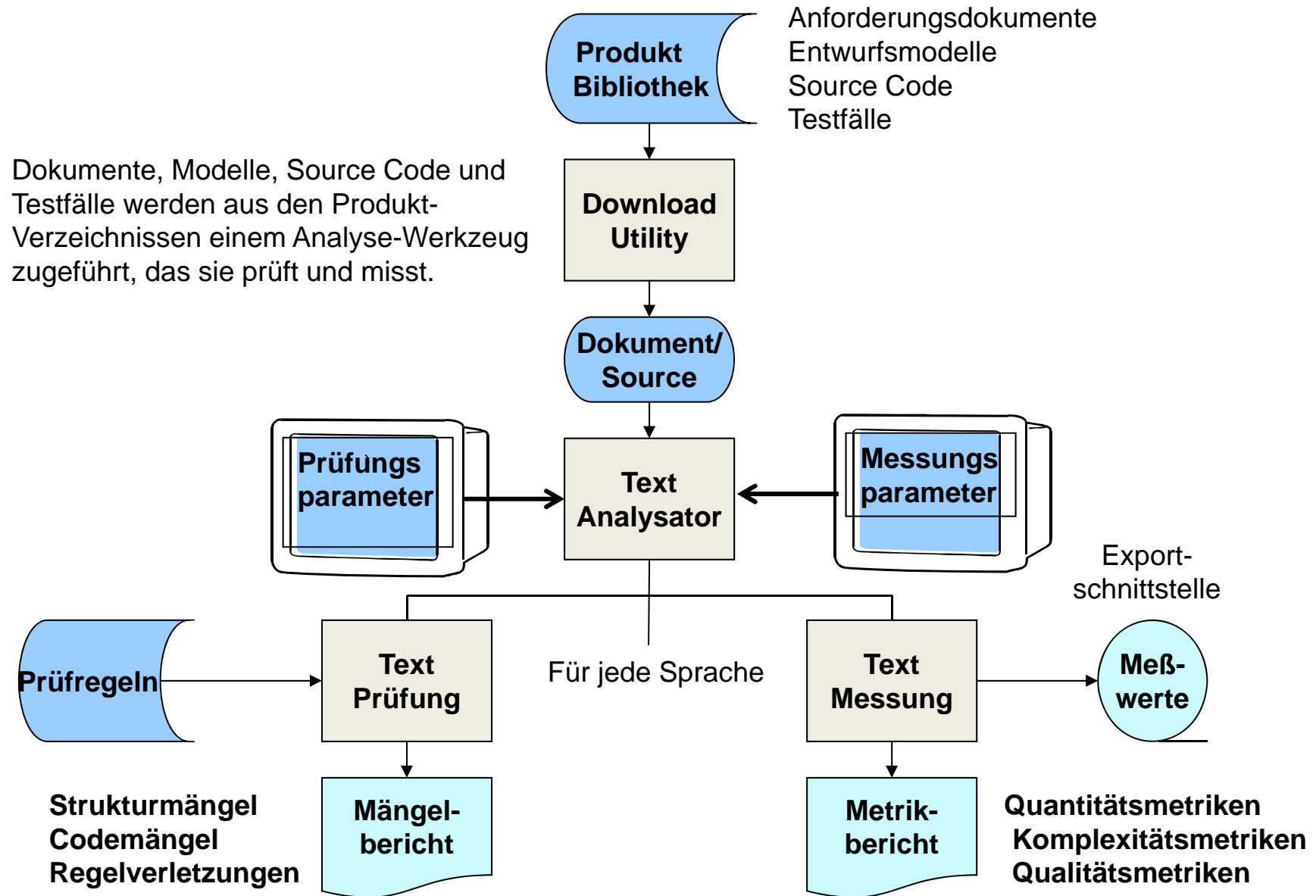
- \* **Dynamische Analyse**
- \* **Modultest**
- \* **Klassentest**
- \* **Programmtest**
- \* **Integrationstest**
- \* **Systemtest**
- \* **Streßtest**
- \* **Funktionstest**
- \* **Abnahmetest**

# Objekte der analytischen Software-Qualitätssicherung



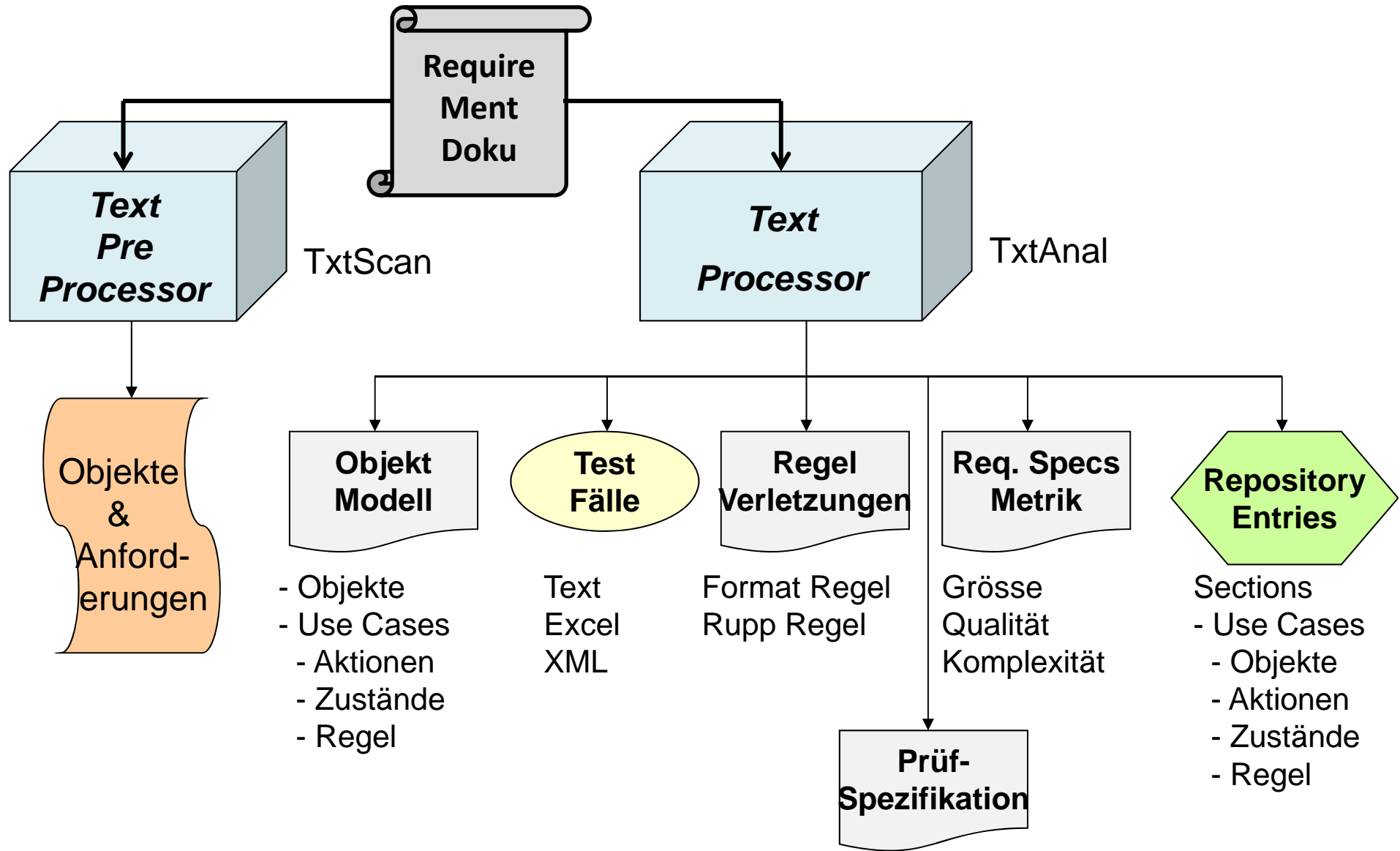
Es komme nicht darauf an wie die Objekte zustande kommen  
Sondern nur dass sie zustandekommen (only the Results count!)

# Methodik der statischen Analyse





# Automatisierte Anforderungsanalyse



Nach V-Modell-XT

# Regeln für die Anforderungsbeschreibung

- 01) Dokument sollte mit einem Titel beginnen.
- 02) Dokument sollte in Abschnitte aufgeteilt sein.
- 03) Dokument sollte erkennbare Anforderungen enthalten.
- 04) Dokument sollte erkennbare Anwendungsfälle enthalten.
- 06) Anforderung sollte mindestens einen Satz haben.
- 07) Anwendungsfall sollte mindestens drei Attribute haben.
- 09) Sätze sollten nicht zu lange sein.
- 11) Anforderungen sollten ausreichend spezifiziert sein.
- 12) Anwendungsfälle sollten ausreichend spezifiziert sein.
- 13) Anforderungen sollten entweder Aktion, Zustand oder Bedingung haben.
- 14) Anwendungsfälle sollten entweder Aktion, Zustand oder Bedingung haben.
- 15) Abschnitte sollten entweder Aktion, Zustand oder Bedingung haben.
- 16) Anforderung sollte sich auf mindestens ein Objekt beziehen.
- 17) Anwendungsfall sollte sich auf mindestens ein Objekt beziehen.
- 18) Abschnitt sollte sich auf mindestens ein Objekt beziehen.
- 19) Jedes definierte Attribut sollte mindestens ein Mal vorkommen.
- 20) Jeder Anwendungsfall sollte eine Vorbedingung haben.
- 21) Jeder Anwendungsfall sollte eine Nachbedingung haben.
- 22) Jeder Anwendungsfall sollte eine Ausnahmebehandlung haben.
- 23) Jeder Anwendungsfall sollte einen Auslöser haben.
- 24) Abkürzungen wie bzw., ggf., usw. sind zu vermeiden.
- 25) Nicht funktionale Anforderungen sollten quantifiziert sein.
- 30) .....
- 31) RUPP: Aktionen sollten im aktiven Modus ausgedruckt werden.
- 32) RUPP: Aktionen sollten mit einem vollen Verb ausgedruckt werden.
- 33) RUPP: Aktionen sollten einen Subjekt und mindestens ein Objekt haben.
- 34) RUPP: Zustand sollte sich auf nur ein Objekt beziehen.
- 35) RUPP: Mögliche und unmögliche Zustände sollten unterschieden werden.
- 36) RUPP: Modale Operatoren sollten eine alternative Aktion vorsehen.
- 37) RUPP: Aussagen sollten keine implizite Annahmen voraussetzen.
- 38) RUPP: Abstrakte Zustände sollten eindeutig definiert sein.
- 39) RUPP: Es soll klar sein welches Objekt das Ziel der Handlung ist.
- 40) RUPP: Verwendete Objekte sollten qualifiziert sein.

# Beispiel der Anforderungsregelprüfung

Aus der Analyse des Lastenheftes für die Nordsee Baggerüberwachung

0086	MINOR	UseCase = Baggerauftragsverwaltung
	RUPP	Die so erstellten Auftraege werden dann per eMail an die Hopperbagger gesendet.
	31	Aktion sollte im aktiven Modus ausgedruckt werden! (wer macht es??)
0087	MINOR	UseCase = Baggerauftragsverwaltung
	RUPP	Die Information des Auftrages wird einerseits an Bord von der Bauaufsicht geprüft
	37	Aussage stellt eine implizite Annahme voraus!
0088	MINOR	UseCase = Baggerauftragsverwaltung
	RUPP	Auftragsnummer ist automatisch zu generieren.
	33	Aktion sollte einen Subjekt haben, z.B. System oder Nutzer (von wem?)
0089	MINOR	UseCase = Baggerauftragsverwaltung
	RUPP	Sie ist an einen Peilplan gebunden, pro Peilplan kann es x Baggerfelder geben.
	40	Zielobjekt sollte qualifiziert sein! Welche Baggerfelder??
0090	MINOR	UseCase = Baggerauftragsverwaltung
	RUPP	Bereichsauftragsnummer wird automatisch generiert
	33	Aktion sollte einen Subjekt haben, z.B. System oder Nutzer (von wem?)
0091	MEDIA	UseCase = Baggerauftragsverwaltung
	REQU	Auftragsdatumserstellung
	13	Anforderung hat weder Zustaende, noch Aktionen, noch Regel (unvollständig)

## Beispiel der Anforderungsmetrik aus dem Lastenheft „Baggerverwaltung“

Number of Requirements specified	=====>	132
Number of User Interfaces specified	=====>	59
Number of User Reports specified	=====>	20
Number of Objects specified	=====>	413
Number of Objects referenced	=====>	1440
Number of Predicates specified	=====>	154
Number of States identified	=====>	117
Number of Actions identified	=====>	186
Number of Rules identified	=====>	136
Number of System Actors specified	=====>	5
Number of Use Cases specified	=====>	31
Number of Use Case Triggers specified	=====>	17
Number of Use Case Paths specified	=====>	62
Number of Use Case Steps specified	=====>	207
Number of UseCase Preconditions specified	=====>	30
Number of UseCase Postconditions specified	=====>	60
Number of UseCase Exceptions specified	=====>	13
Number of UseCase Relations specified	=====>	28
Number of UseCase Attributes specified	=====>	788
Number of Test Cases extracted	=====>	407

### L A S T E N H E F T      S I Z E      M E T R I C S

Number of Function-Points	=====>	1084
Number of Data-Points	=====>	4288
Number of Object-Points	=====>	4938
Number of Use Case Points	=====>	480
Number of Test Case Points	=====>	683

# Beispiel der Anforderungsmetrik

REQUIREMENT	SIZE	METRICS
Number of Function-Points		=====> 188
Number of Data-Points		=====> 961
Number of Object-Points		=====> 726
Number of Use Case Points		=====> 108
Number of Test Case Points		=====> 255

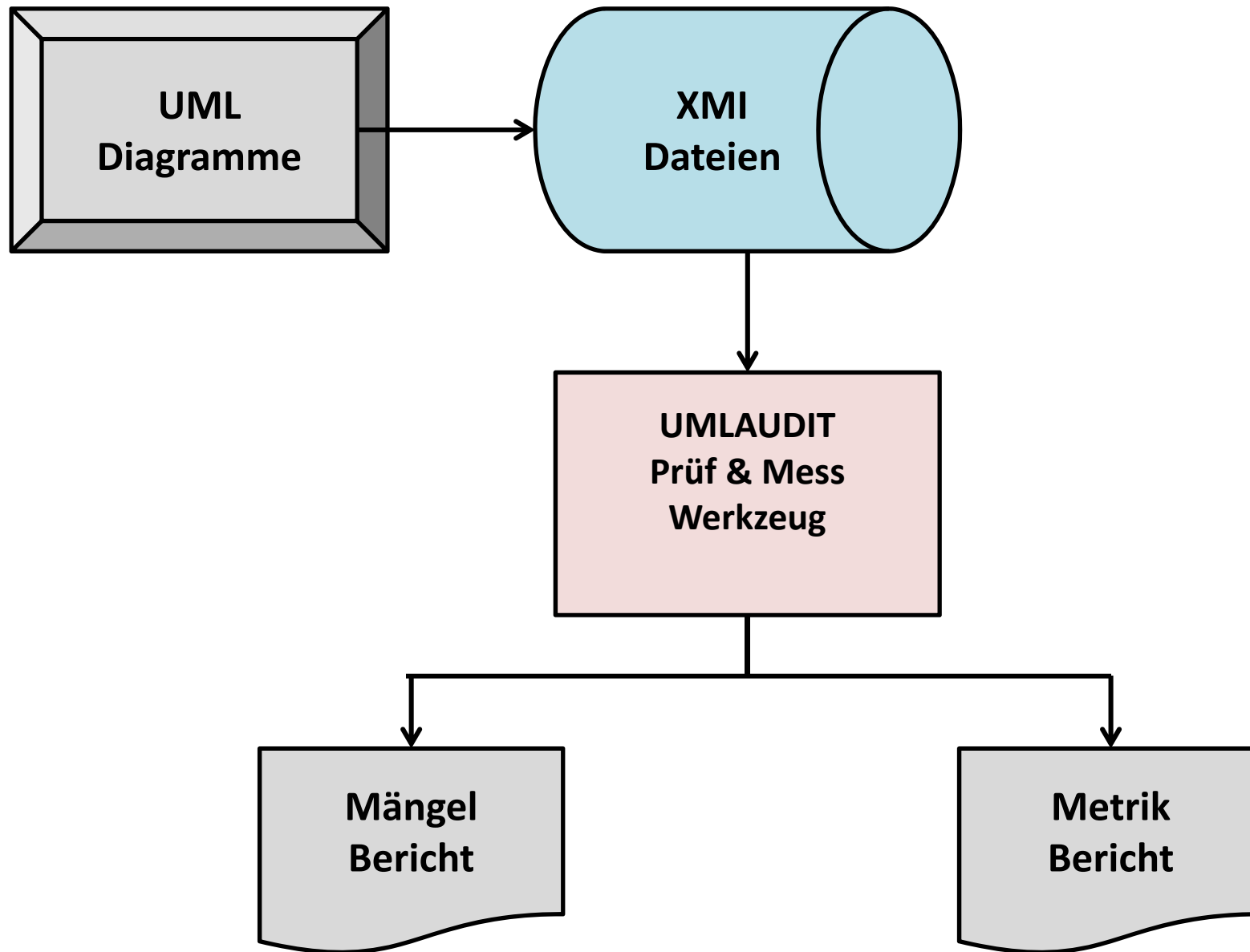
---

REQUIREMENT	COMPLEXITY	METRICS
Data Density		=====> 0.236
Functional Density		=====> 0.100
State Density		=====> 0.386
Conditional Density		=====> 0.402
Referential Density		=====> 0.768
Test Case Density		=====> 0.440
Overall Requirement Complexity Rating		=====> 0.388

---

REQUIREMENT	QUALITY	METRICS
Degree of Completeness		=====> 0.966
Degree of Consistency		=====> 0.874
Degree of Stability		=====> 0.896
Degree of Changeability		=====> 0.678
Degree of Testability		=====> 0.364
Degree of Conformity		=====> 0.941
Overall Requirement Quality Rating		=====> 0.786

# Automatisierte Analyse des Design-Modells



LEVEL	UML-TYPE	UML ENTITY NAME / DEFICIENCY MESSAGE
Design	NAME	cqi
MINOR	Rule:39	Entity Name does not conform to naming rules!
Design	NAME	aos
MINOR	Rule:39	Entity Name does not conform to naming rules!
Design	CLASS	category
MAJOR	Rule:06	Required Business Object is not in UML Model!
Design	CLASS	displayErrorMessage
MINOR	Rule:31	Class referenced is not defined!
Design	OBJECT	DISPATCH_ORDER_ITEM
MAJOR	Rule:06	Required Business Object is not in UML Model!
Design	USECASE	USECASE_01_CUSTOMER_QUERY
MAJOR	Rule:07	Required Use Case is not in the UML Model!
Design	USECASE	USECASE_02_CUSTOMER_ORDER_PROCESSING
MAJOR	Rule:07	Required Use Case is not in the UML Model!

# Design-Metrikbericht (Entities)

Q U A N T I T Y     M E T R I C S		
D E S I G N	D I A G R A M	M E T R I C S
Number of Design Diagrams analyzed	=====>	27
Number of UseCase Diagrams analyzed	=====>	1
Number of Activity Diagrams analyzed	=====>	6
Number of Class Diagrams analyzed	=====>	11
Number of Sequence Diagrams analyzed	=====>	6
Number of State Diagrams analyzed	=====>	2
Number of Component Diagrams analyzed	=====>	2
S T R U C T U R A L   Q U A N T I T Y   M E T R I C S		
Number of Design Entities	=====>	876
Number of Design Entities referenced	=====>	653
Number of Systems designed	=====>	1
Number of Business Objects required	=====>	20
Number of Business Objects specified	=====>	24
Number of Use Cases required	=====>	6
Number of Use Cases specified	=====>	6
Number of System Actors required	=====>	5
Number of System Actors specified	=====>	6
Number of System Components specified	=====>	2
Number of Class Interfaces specified	=====>	37
Number of Classes specified	=====>	24
Number of Base/Super Classes specified	=====>	6
Number of Methods specified	=====>	81



# Design Metrikbericht (Relationships)

## RELATIONAL QUANTITY METRICS

Number of Design Relationships	=====>	712
Number of Use Case Relations	=====>	394
Number of Class Associations	=====>	24
Number of Class Generalizations	=====>	16
Number of Class Hierarchy Levels	=====>	4
Number of Interactions/Collaborations	=====>	88
Number of Methods referenced	=====>	38
Number of Activity Control Flows	=====>	173
Number of State Transitions	=====>	17
Number of required Test Cases	=====>	756

## DESIGN SIZE METRICS

Number of Data-Points	=====>	358
Number of Object Points	=====>	773
Number of Function Points	=====>	273
Number of Use-Case Points	=====>	70
Number of Test Points	=====>	1187

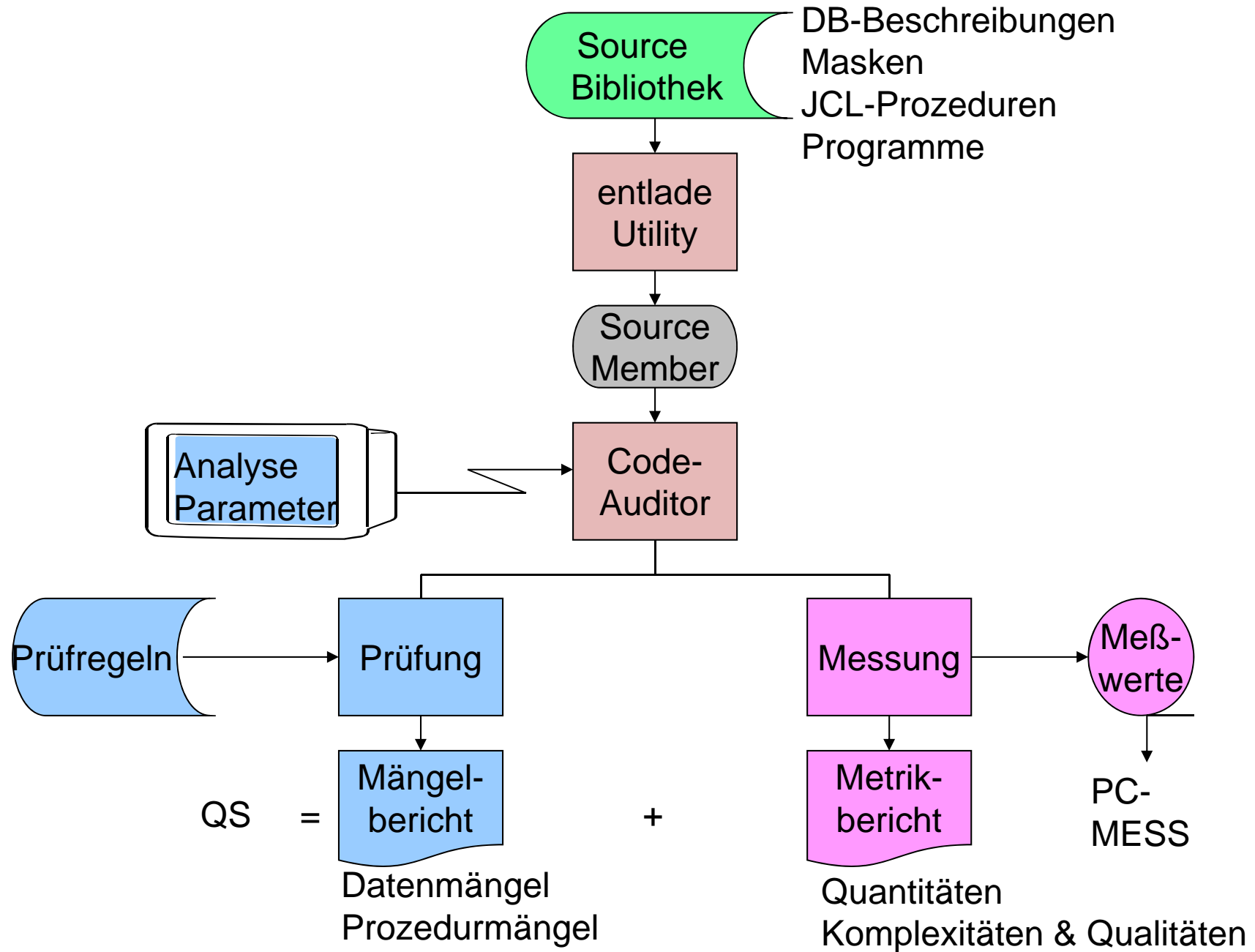
**D E S I G N   C O M P L E X I T Y   M E T R I C S**

CLASS INTERACTION COMPLEXITY	=====>	0.591
CLASS HIERARCHICAL COMPLEXITY	=====>	0.208
CLASS DATA COMPLEXITY	=====>	0.813
CLASS FUNCTIONAL COMPLEXITY	=====>	0.407
STATE COMPLEXITY	=====>	0.941
STATE TRANSITION COMPLEXITY	=====>	0.235
ACTIVITY COMPLEXITY	=====>	0.780
ACTOR INTERACTION COMPLEXITY	=====>	0.166
OVERALL DESIGN COMPLEXITY	=====>	0.448
AVERAGE DESIGN COMPLEXITY	=====>	0.508

**D E S I G N   Q U A L I T Y   M E T R I C S**

CLASS COUPLING	=====>	0.788
CLASS COHESION	=====>	0.313
DESIGN MODULARITY	=====>	0.888
DESIGN PORTABILITY	=====>	0.426
DESIGN REUSABILITY	=====>	0.388
DESIGN TESTABILITY	=====>	0.629
DESIGN COMPLETENESS	=====>	0.500
DESIGN CONSISTENCY	=====>	0.735
DESIGN COMPLIANCE	=====>	0.960
AVERAGE DESIGN QUALITY	=====>	0.564

# Automatisierte Source-Code Analyse



## Aufdeckung von Codemängeln

Es gibt drei Möglichkeiten Codemängel aufzudecken:

- **Code Inspection** = Durchlesen des Codes mit einer Checkliste
- **Dynamic Analysis** = kontrollierte Ausführung des Codes
- **Static Analysis** = automatische Analyse des Source Codes

Feldexperimente haben gezeigt, dass viele Fehler = defects (> 70%) und fast alle Mängel = Deficiencies durch Code Inspektion oder statische Analyse aufgedeckt werden können.

Statische Analyse, bzw. Code-Auditing, ist der schnellste und billigste Weg Fehler und Mängel zu finden.

## Codemängelklassen

Aus der Sicht der Softwarequalität gibt es vier Klassen von Codemängel:

- **Fehlender Code** = Anweisungen, die vorhanden sein müssten und die abwesend sind. (Gaps)
- **Redundanter Code** = Anweisungen, die vielfach vorkommen, wenn einmal würde genügen. (Clones)
- **Fehlerhafter Code** = Anweisungen, die einen Abbruch oder ein falsches Ergebnis erzeugen wenn sie ausgeführt werden. (Defects)
- **Non-konformer Code** = Anweisungen die eine formale Codierregel verletzen. (Rule Violations)
- **Riechender code** = Anweisungen, welche die Fortschreibung des Codes beeinträchtigen. (Code Smells)

## Beispiele von Sicherheitsmängeln

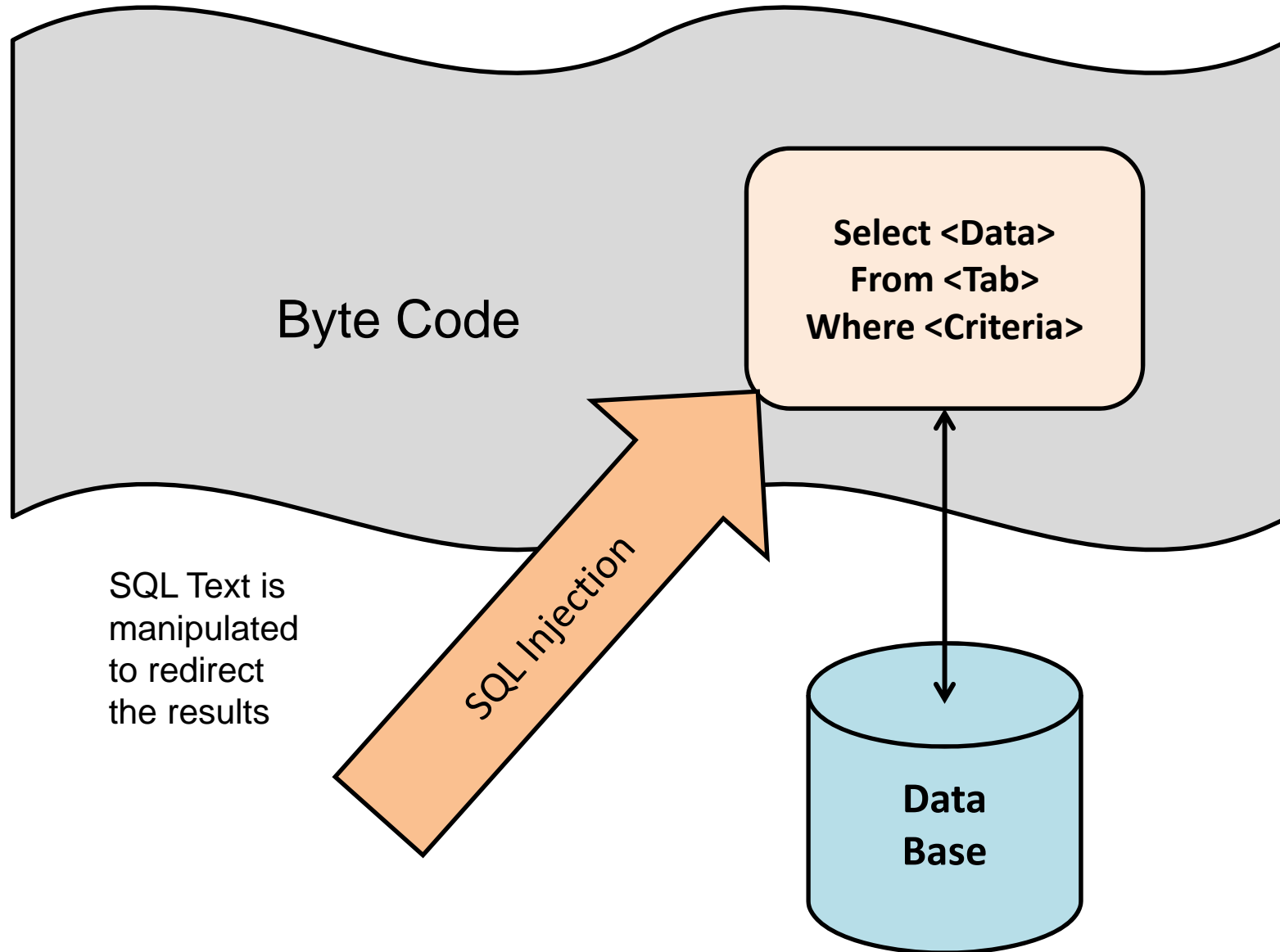
- **Cross-Site Scripting**
- **Hard-coded Data**
- **Embedded SQL statements in the code**
- **Nested Classes**
- **Cloned Classes & Methods**
- **Non-Final Classes & Methods**
- **Non-deleted Objects**
- **Non-matching interfaces**
- **Uncontrolled return values**
- **Unsecured Reflection**
- **Unchecked Inputs**

# Hard-coded Data

```
public static String getBankverbindung(KundenLandEnum KundenLand)
{
    String bankVerbindung;
    switch (KundenLand)
    {
        case KundenLandEnum.Oberösterreich:
            bankVerbindung = "Bankverbindung: RZB Linz BLZ 15000 KtNr. 420 5142 00";
            break;
        case KundenLandEnum.Kärnten:
            bankVerbindung = "PSK, 60000 2339078";
            break;
        case KundenLandEnum.Tirol:
            bankVerbindung = „TirolerLandesbank BLZ 19000 KtNr. 356 0983-299“;
            break;

        default:
            bankVerbindung = “ ”;
            break;
    }
    return bankVerbindung;
}
/* getBankverbindung */
```

# Embedded SQL Operationen





# Sample SQL Injection

SQL Queries can be manipulated

Exposed SQL Command

```
public Vector listAnschriften() throws DatenbankException
{
    try {
        Vector vAnschriften = new Vector();
        Statement s = transaktion.getStatement();

        ResultSet r = s.executeQuery("SELECT Anschriftnummer
                                    FROM Kundenanschrift WHERE Kundennummer=" + iNr
                                    + " ORDER BY Anschriftnummer");

        while (r.next())
            vAnschriften.addElement(new Anschrift(r.getInt("Anschriftnummer")));
        s.close();
        return vAnschriften;
    } catch (Exception fehler)
    {
        throw new DatenbankException
            ("Kann Anschriften von Kunde " + iNr + " nicht auflisten.", fehler);
    }
} /**
```

```
s.executeQuery("SELECT Anschriftnummer
               FROM Kundenanschrift WHERE Kundennummer=" + 4711);
```

## Private Classes nested in Public Classes

```
Package mypackage

Public final class EnclosingClass {

    // private nested class
    private static final class NestedClass {
        private int nestedInt = 10;
    }

    Public static void main{String[ ] args) {
        EnclosingClass.NestedClass nested = new EnclosingClass.NestedClass();
        System.out.println ( "nestedInt = " + nested.nestedInt);
    }
}
```

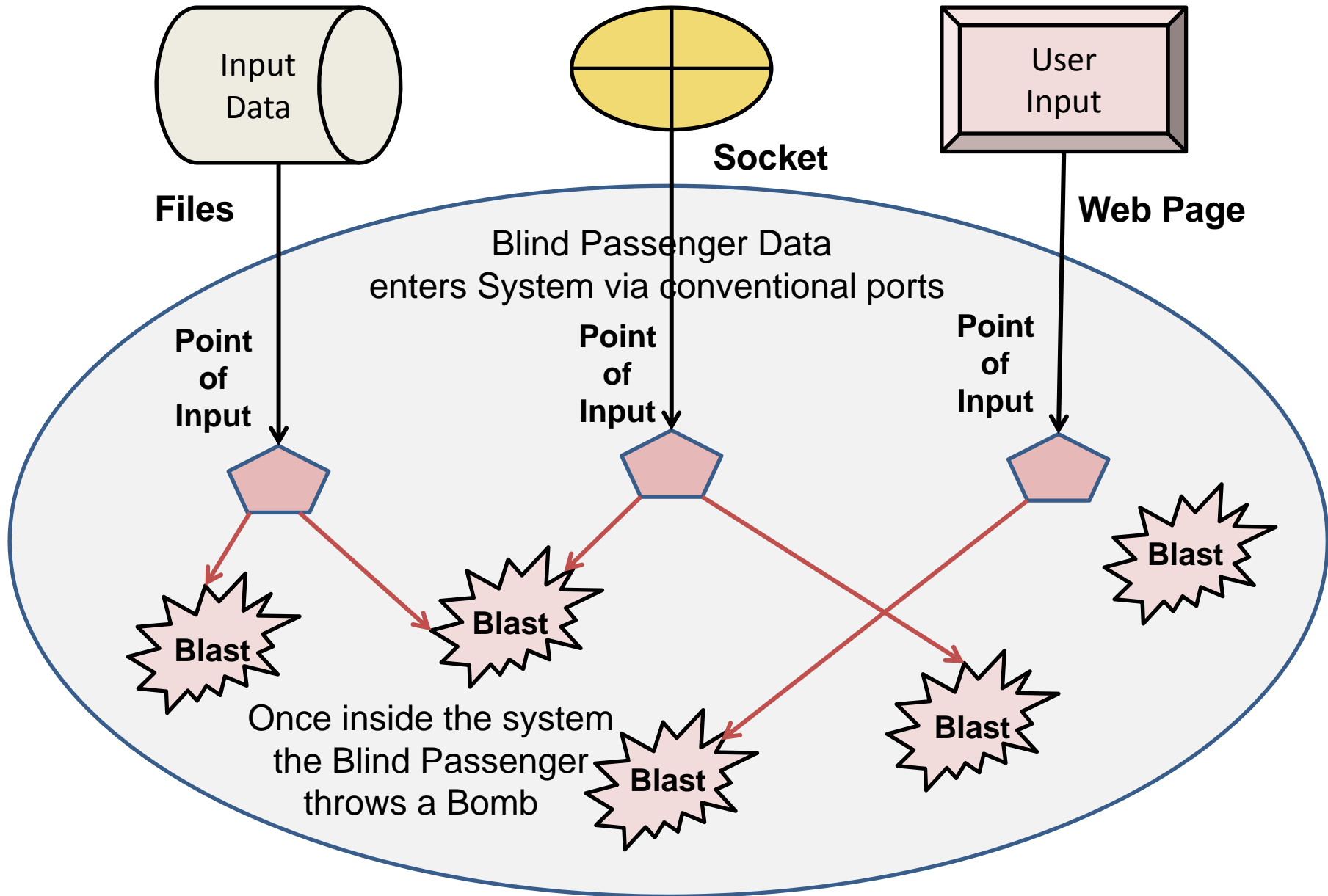
nestedInt was intended to be a private variable but since it's class is Nested in a public class, it too becomes public. This can have consequences For the security since than foreign components can access the nestedInt.

## Non-Matching Parameterlisten

```
// Parameter exception Error
interface StatusReport{
    void ShowHeight (double amount);
    void ShowSpeed (double speed);
}
class Jet implements StatusReport{
    double height; int speed;
    Jet (double h, int s) {
        height = h; speed = s;
    }
    public void ShowHeight (double h) {
        system.out.Println("Flying at "+h+".");
    }
}
// Instantiation of an Object of Type Jet
class JumpJet {
    public static void main (String [] args) {
        Jet vtol = new Jet(5000, 550);
        vtol.height = vtol.height - 50;
        vtol.ShowHeight(vtol.Height);
    }
}
```

Later the interface StatusReport is altered to include a new method ShowSpeed. Adding a method to an interface is a binary compatible change with no effect until Jet is recompiled. Then the program will break since Jet has no ShowSpeed implementation.

# Nicht geprüfte Eingangsdaten



# Erkennung eingebetteter SQL Operationen

SoftRepo - Database : C:\HARRY\PROJECTS\MEASUREMENT\ZTP\OUTPUTS\PHP\DeficiencyRepository\

Database operations Tree operations Select view Outputs Print Help

Import data file(s) to the repository Select a root element View repository content Edit repository content \sysext\frontend\Classes\Plugin\AbstractPlugin.php

Node details

Error Code	68101	Source/Line
PHP-50	found_in	AbstractPlugin.php:1041
		SQL Select statements are insecure

Apply changes

Error - Classes

- is
  - Error Class - Deficiency
  - Error Class - Problem
  - Error Class - Security violation
    - is
      - Error Code - PHP-05 : Derived Class is not declared final
      - Error Code - PHP-11 : Input Parameters should be checked before they are used!
      - Error Code - PHP-16 : Return Value is not controlled after function call!
      - Error Code - PHP-49 : Embedded SQL should be avoided!
      - Error Code - PHP-50 : SQL Select statements are insecure!
        - found\_in
        - FNDS
      - Error Code - PHP-67 : SQL Search Clauses with OR condition are dangerous
- Error Class - Warning

```

}
// Search word:
if ($this->piVars['sword'] && $this->internal['searchFieldList']) {
    $WHERE = $this->cObj->searchWhere($this->piVars['sword'], $this->internal['searchFieldList']);
}
if ($count) {
    $queryParts = array(
        'SELECT' => 'count(*)',
        'FROM' => $TABLENAMES,
        'WHERE' => $WHERE,
        'GROUPBY' => '',
        'ORDERBY' => '',
        'LIMIT' => ''
    );
} else {
    // Order by data:
    if (!$orderBy && $this->internal['orderBy']) {
        if (GeneralUtility::inList($this->internal['orderByList'], $this->internal['orderBy'])) {
            $orderBy = 'ORDER BY ' . $table . '.' . $this->internal['orderBy'] . ' ';
        }
    }
    // Limit data:
    $pointer = (int)$this->piVars['pointer'];
    $results_at_a_time = \TYPO3\CMS\Core\Utility\MathUtility::forceIntegerInRange($this->piVars['results_at_a_time'], 1, 100, 10);
    $LIMIT = $pointer * $results_at_a_time . ' ' . $results_at_a_time;
    // Add 'SELECT'
    $queryParts = array(
        'SELECT' => $this->pi_prependFieldsWithTable($table, $this->pi_listFields($table)),
        'FROM' => $TABLENAMES,
        'WHERE' => $WHERE,
        'GROUPBY' => $GLOBALS['TYPO3_DB']->stripGroupBy($groupBy),
        'ORDERBY' => $GLOBALS['TYPO3_DB']->stripOrderBy($orderBy),
        'LIMIT' => $LIMIT
    );
}
return $GLOBALS['TYPO3_DB']->exec_SELECT_queryArray($queryParts);

```

Line 1041 of 1: AbstractPlugin.php

# Erkennung von Data Casting in PHP Code

SoftRepo - Database : C:\HARRY\PROJECTS\MEASUREMENT\ZTP\OUTPUTS\PHP\DeficiencyRepository\

Database operations Tree operations Select view Outputs Print Help

Bottom reached, no more forward references!

Import data file(s) to the repository Select a root element View repository content Edit repository content \sysext\install\Classes\Controller\Action\Ajax\ExtensionCompatibilityTester.php

Node details

Error Code	38218	Source/Line
PHP-18	found_in	ExtensionCompatibilityTester.php:00117

Data type casting should be avoided! Apply changes

Source/Line - AlphanumericFilter.php:00061 : Data type casting should be avoided!!

Source/Line - BackendUtility.php:01404 : Data type casting should be avoided!!

Source/Line - ConfigurationForm.php:00181 : Data type casting should be avoided!!

Source/Line - DigitFilter.php:00033 : Data type casting should be avoided!!

Source/Line - ExtensionCompatibilityTester.php:00117 : Data type casting should be avoided!!

Source/Line - GeneralUtility.php:00954 : Data type casting should be avoided!!

Source/Line - GeneralUtility.php:00977 : Data type casting should be avoided!!

Source/Line - GeneralUtility.php:00981 : Data type casting should be avoided!!

Source/Line - GeneralUtility.php:00985 : Data type casting should be avoided!!

Source/Line - HtmlParser.php:01436 : Data type casting should be avoided!!

Source/Line - RegExpFilter.php:00058 : Data type casting should be avoided!!

Source/Line - RteHtmlParser.php:01617 : Data type casting should be avoided!!

Source/Line - StringUtility.php:00048 : Data type casting should be avoided!!

Source/Line - adodb-active-record.inc.php:00648 : Data type casting should be avoided!!

Source/Line - adodb-active-recordx.inc.php:00720 : Data type casting should be avoided!!

Source/Line - adodb-ibase.inc.php:00613 : Data type casting should be avoided!!

Source/Line - adodb-ibase.inc.php:00654 : Data type casting should be avoided!!

Source/Line - adodb-ibase.inc.php:00664 : Data type casting should be avoided!!

Source/Line - adodb-informix72.inc.php:00288 : Data type casting should be avoided!!

Source/Line - adodb-oci8.inc.php:00310 : Data type casting should be avoided!!

Source/Line - adodb-oracle.inc.php:00059 : Data type casting should be avoided!!

Source/Line - adodb-oracle.inc.php:00067 : Data type casting should be avoided!!

Source/Line - adodb-time.inc.php:01031 : Data type casting should be avoided!!

Source/Line - adodb-time.inc.php:01033 : Data type casting should be avoided!!

Source/Line - adodb-xmlschema.inc.php:01731 : Data type casting should be avoided!!

Source/Line - adodb.inc.php:02520 : Data type casting should be avoided!!

Source/Line - adodb.inc.php:02528 : Data type casting should be avoided!!

FNDS

Error Code - PHP-20 : Multidimensional Arrays are not allowed!

```

*
* @return array
*/
protected function getExtensionsToExclude() {
    $exclude = Utility\GeneralUtility::getUrl($this->protocolFile);
    return Utility\GeneralUtility::trimExplode(',', (string)$exclude);
}

/**
 * Tries to load the ext_localconf and ext_tables files of all non-core extensions
 * Writes current extension name to file and deletes it again when inclusion was
 * successful.
 *
 * @param array $extensions
 * @return void
 */
protected function tryToLoadExtLocalconfAndExtTablesOfExtensions(array $extensions) {
    foreach ($extensions as $extensionKey => $extension) {
        $this->writeCurrentExtensionToFile($extensionKey);
        $this->loadExtLocalconfForExtension($extensionKey, $extension);
        $this->removeCurrentExtensionFromFile($extensionKey);
    }
    Utility\ExtensionManagementUtility::loadBaseTca(FALSE);
    foreach ($extensions as $extensionKey => $extension) {
        $this->writeCurrentExtensionToFile($extensionKey);
        $this->loadExtTablesForExtension($extensionKey, $extension);
        $this->removeCurrentExtensionFromFile($extensionKey);
    }
}

/**
 * Loads ext_tables.php for a single extension. Method is a modified copy of
 * the original bootstrap method.
 *
 * @param string $extensionKey
 * @param \ArrayAccess $extension
 * @return void
 */
protected function loadExtTablesForExtension($extensionKey, array $extension) {
    // In general it is recommended to not rely on it to be globally defined in the
    // scope, but we can not prohibit this without breaking backwards compatibility
    global $T3_SERVICES, $T3_VAR, $TYPO3_CONF_VARS;
    global $TBE_MODULES, $TBE_MODULES_EXT, $TCA;
    global $PAGES_TYPES, $TBE_STYLES, $FILEICONS;
    global $_EXTKEY;
    // Load each ext_tables.php file of loaded extensions
    $_EXTKEY = $extensionKey;
    if (isset($extension['ext_tables.php']) || $extension['ext_tables.php']) {

```

Line 117 of 25 | ExtensionCompatibilityTester.php

# Fehlende Prüfung der Eingangsdaten in PHP Code

SofAudit main shell - PHP [Tpos-3PhsysPhp] Last batch runtime: 01:45:04:899

File Actions Viewers/Editors Logs Help

Language selection Names and datasets Analysis parameters Internal viewer Edit word list

Defis\sysext\backend\AbstractRecordList.php.DEF

Component: sysext Module: backend Source: AbstractRecordList.php

Sort by source file name Sort by rules Sort by deficiency level Print Close

Number of Secure Rule Violations= 6  
 Number of major Rule Violations = 16  
 Number of medium Rule Violations= 17  
 Number of minor Rule Violations = 12  
 Total Number of Rule Violations = 51  
 Number of Source Statements = 108  
 Rate of Rule Conformity = 0.453

AbstractRecordList.php

- Security violations
  - 11 Input Parameters should be checked before
    - 193
    - 272
    - 326**
    - 349
    - 362
    - 399
- Problems
  - 01 Abstract Class should be defined as Interface
  - 15 Functions should not be returned from another function
  - 20 Multidimensional Arrays are not allowed!
  - 29 \$ sign should not appear within a literal string
  - 20 Multidimensional Arrays are not allowed!
  - 15 Functions should not be returned from another function
  - 26 Default is missing in last Switch Statement!
  - 15 Functions should not be returned from another function
  - 07 Scope definition is missing for this function!
  - 17 Constants should not be passed as parameter
  - 07 Scope definition is missing for this function!
  - 20 Multidimensional Arrays are not allowed!
  - 29 \$ sign should not appear within a literal string
  - 12 Class should only contain one public method
- Defis
- Warnings

```

    * @return string
    * @access private
    * @todo Define visibility
    */
    public function fwd_rwd_HTML($type, $pointer, $table = '') {
        $content = '';
        $iParam = $table ? '&table=' . rawurlencode($table) : '';
        switch ($type) {
            case 'fwd':
                $href = $this->listURL() . '&pointer=' . ($pointer - $this->iLimit) . $iParam;
                $content = '<a href="' . htmlspecialchars($href) . '">'. IconUtility::getSpriteIcon('actions-move-up') . '</a>';
                break;
            case 'rwd':
                $href = $this->listURL() . '&pointer=' . $pointer . $iParam;
                $content = '<a href="' . htmlspecialchars($href) . '">'. IconUtility::getSpriteIcon('actions-move-down') . '</a>';
                break;
        }
        return $content;
    }

    /**
     * Creates the URL to this script, including all relevant GPs
     *
     * @param string $altId Alternative id value. Enter blank string for the current id ($this->id)
     * @return string URL
     * @todo Define visibility
     */
    public function listURL($altId = '') {
        return $this->getThisScript() . 'id=' . ($altId !== '' ? $altId : $this->id);
    }

    /**
     * Returning JavaScript for Clipboard functionality.
     *
     * @return string
     * @todo Define visibility
     */
    public function CBfunctions() {
        return '
        // checkOffCB()
        function checkOffCB(listOfCBnames, link) { //
            var checkBoxes, flag, i;
            var checkBoxes = listOfCBnames.split(",");
            if (link.rel === "") {
                link.rel = "allChecked";
            }
        }
    }
  
```

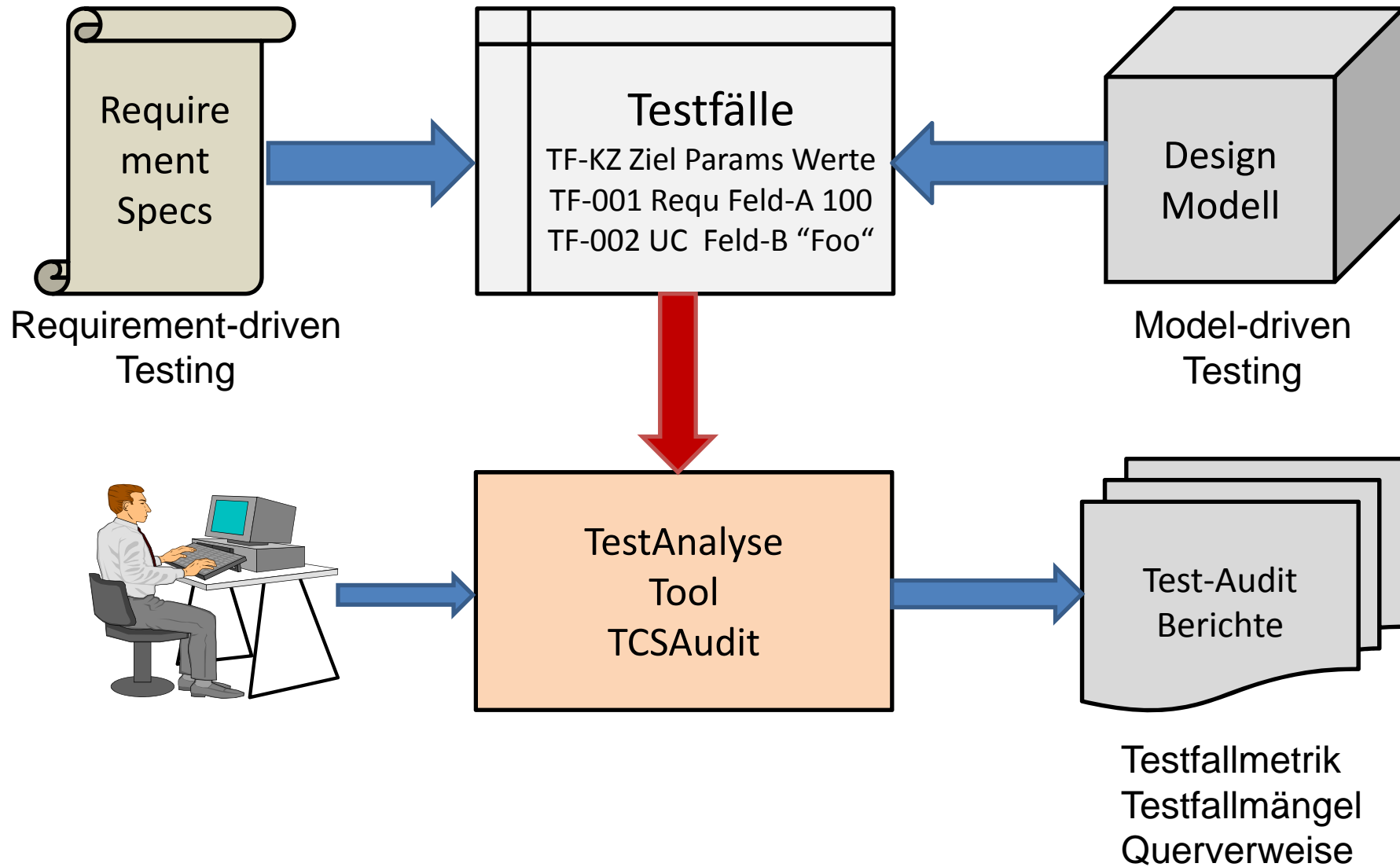
Line 326 of 467 in C:\HARRY\PROJECTS\MEASUREMENT\TYPO3\typo3\_src\typo3\sysext\backend\Classes\RecordList\AbstractRecordL...

# Code-Metrikbericht für Java Code

C O M P L E X I T Y		M E T R I C S	
DATA COMPLEXITY (Chapin Metric)	=====>	0.592	
DATA FLOW COMPLEXITY (Elshof Metric)	=====>	0.792	
DATA ACCESS COMPLEXITY (Card Metric)	=====>	0.920	
INTERFACE COMPLEXITY (Henry Metric)	=====>	0.172	
CONTROL FLOW COMPLEXITY (McCabe Metric)	=====>	0.527	
DECISIONAL COMPLEXITY (McClure Metric)	=====>	0.641	
BRANCHING COMPLEXITY (Sneed Metric)	=====>	0.960	
LANGUAGE COMPLEXITY (Halstead Metric)	=====>	0.556	
WEIGHTED AVERAGE PROGRAM COMPLEXITY	=====>	0.645	
Q U A L I T Y		M E T R I C S	
DEGREE OF MODULARITY	=====>	0.586	
DEGREE OF PORTABILITY	=====>	0.742	
DEGREE OF REUSABILITY	=====>	0.960	
DEGREE OF TESTABILITY	=====>	0.548	
DEGREE OF CONVERTIBILITY	=====>	0.123	
DEGREE OF FLEXIBILITY	=====>	0.535	
DEGREE OF CONFORMITY	=====>	0.488	
DEGREE OF MAINTAINABILITY	=====>	0.444	
WEIGHTED AVERAGE PROGRAM QUALITY	=====>	0.540	



# Testfallqualitätsanalyse



# Testfallmängelbericht

ORDERS0181	RULE	•INTEGRITY = AT LEAST 80% OF ALL BAD INPUT SHOULD BE DETEC
Problem:	Msg:06	Test Case is not assigned to a tester
Missing:	Msg:12	Test Case automation status is missing
Warning:	Msg:14	Test Case source is missing
ORDERS0182	RULE	•PERFORMANCE = THE RESPONSE TIME CONSTRAINT OF THREE SECO
Problem:	Msg:06	Test Case is not assigned to a tester
Warning:	Msg:14	Test Case source is missing
ORDERS0183	RULE	•SECURITY = 90% OF ALL ATTEMPTS TO BREAK INTO THE SYSTEM
Problem:	Msg:04	Test Case is not linked to a Use Case
Problem:	Msg:06	Test Case is not assigned to a tester
Warning:	Msg:14	Test Case source is missing
ORDERS0184	STATE	A TEST COVERAGE OF AT LEAST 80% BRANCH COVERAGE IS REQUI
Problem:	Msg:06	Test Case is not assigned to a tester
Missing:	Msg:12	Test Case automation status is missing
Warning:	Msg:14	Test Case source is missing
ORDERTEST-TEST	Proc	Limits exceeded
Warning:	Msg:19	Max Limit of TestCase Attributes exceeded
-----		
		Number of major Rule Violations = 198
		Number of media Rule Violations = 184
		Number of minor Rule Violations = 189
		Total Number of Rule Violations = 571
		Number of Functional Test Cases = 184
		Number of Test Case Attributes = 2944
		Rate of Attribute Conformity = 0.804
-----		

# Testfallkonsistenzbericht

Requirement Name	in Document Section	TestCase Name
FUNC-REQ-05_Resupplying	OrderEntry	ORDERS0016
FUNC-REQ-05_Resupplying	OrderEntry	ORDERS0017
FUNC-REQ-05_Resupplying	OrderEntry	ORDERS0018
FUNC-REQ-05_Resupplying	OrderEntry	ORDERS0019
FUNC-REQ-05_Resupplying	OrderEntry	ORDERS0020
FUNC-REQ-06_BackOrderProcessing	OrderEntry	ORDERS0022
FUNC-REQ-06_BackOrderProcessing	OrderEntry	ORDERS0023
FUNC-REQ-06_BackOrderProcessing	OrderEntry	ORDERS0024
NF-REQ-01_ReponseTime	OrderEntry	Testcase is missing!
NF-REQ-02_TransactionCapacity	OrderEntry	ORDERS0027
NF-REQ-03_Availability	OrderEntry	ORDERS0028
NF-REQ-04_Security	OrderEntry	ORDERS0029
NF-REQ-04_Security	OrderEntry	ORDERS0030
NF-REQ-04_Security	OrderEntry	ORDERS0031
NF-REQ-05_Recoverability	OrderEntry	ORDERS0032
NF-REQ-07_Usability	OrderEntry	ORDERS0035
Number of Requirements to be tested	=	20
Number of Requirements with a Test Case	=	19
Number of Requirements without a Test Case	=	1
Degree of Requirements Coverage	=	0.95

# Testfallmetrikbericht (Größen)

T C S A N A L T E S T C A S E M E T R I C R E P O R T		
T E S T C A S E Q U A N T I T Y		M E T R I C S
Number of System Modules analyzed	=====>	2
Number of System Documents analyzed	=====>	2
Number of Test Procedures analyzed	=====>	4
Number of Text Cases analyzed	=====>	368
Number of Test Case Attributes specified	=====>	5888
Number of Requirements to be tested	=====>	40
Number of Use Cases to be tested	=====>	12
Number of Data Objects to be tested	=====>	346
Number of Requirements tested	=====>	38
Number of Use Cases tested	=====>	12
Number of Data Objects tested	=====>	346
Number of Actions to be tested	=====>	54
Number of States to be tested	=====>	120
Number of Rules to be tested	=====>	194
Number of different Test Case types	=====>	6
Number of Test Case sources	=====>	0
Number of automated Test Cases	=====>	0
Number of Triggers specified	=====>	368
Number of Preconditions specified	=====>	368
Number of Postconditions specified	=====>	368
T E S T C A S E S I Z E		M E T R I C S
Number of Test Points	=====>	376
Number of Data Points	=====>	1384

# Testfallmetrikbericht (Komplexität & Qualität)

T C S A N A L T E S T C A S E M E T R I C R E P O R T		
T E S T C A S E C O M P L E X I T Y M E T R I C S		
Test Case Complexity Ratio	=====>	0.280
Test Case Density Ratio	=====>	0.920
Test Case Intensity Ratio	=====>	0.859
Test Case Volume Ratio	=====>	0.750
Overall Test Case Complexity Rating	=====>	0.652
T E S T C A S E Q U A L I T Y M E T R I C S		
Test Case Maintainability Ratio	=====>	0.940
Test Case Coverage Ratio	=====>	0.961
Test Case Completeness Ratio	=====>	0.805
Test Case Reusability Ratio	=====>	0.080
Overall Test Case Quality Rating	=====>	0.696
T E S T C A S E D E F I C I E N C Y M E T R I C S		
Number of Major Rule Violations	=====>	396
Number of Medium Rule Violations	=====>	368
Number of Minor Rule Violations	=====>	378
Number of Missing Attributes	=====>	744
Number of Missing References	=====>	396

# Konsistenzprüfung über alle semantische Ebenen hinaus

SYSTEM CONSISTENCY METRIC REPORT			
LANGUAGE: GERMAN/UML/C++/TCS		DATE: 22.06.13	
SYSTEM: LAGERHALTUNG		PAGE: 3 of 19	
ENTITY COUNTS			
Number of System TestCases	=====>	356	
Number of Requirements to be tested	=====>	151	
Number of Requirements with TestCases	=====>	124	
Number of Code Components to be tested	=====>	229	
Number of Code Components with TestCases	=====>	80	
RELATION COUNTS			
Number of TestCase/Requirement Relations	=====>	699	
Number of TestCase/Component Relations	=====>	10086	
DEFICIENCY COUNTS			
Number of Requirements with no TestCase	=====>	27	
Number of Code Components with no TestCase	=====>	149	
COVERAGE METRICS			
Requirement	Test Coverage Rate	=====>	0.821
Code Component	Test Coverage Rate	=====>	0.349

# Wozu das denn Alles?

## Hat Qualitätssicherung überhaupt einen Sinn?

- Qualitätsprüfungen könnten als Lernhilfe für die Entwickler dienen.
- Qualitätsprüfungen mahnen die Entwickler auf dem richtigen Pfad zu bleiben.
- Andererseits genießen Entwickler Narrenfreiheit, sie können machen was sie wollen.
- Es gebe in der heutigen liberalen anti-autoritären Gesellschaft wenig Möglichkeiten das Verhalten der Entwickler zu beeinflussen.
- Eine Möglichkeit wäre es die Bezahlung mit der Menge und Qualität der gelieferten Ergebnisse zu koppeln.