Saxonia Systems

A vision of technology.

Das literarische Code-Quartett

Immanuel Scholz
immanuel.scholz@saxsys.de

Ralf Ebert
ralf.ebert@saxsys.de

Das literarische Code-Quartett empfiehlt:

Checkliste für schlechten Code

☑ Genügend Codevorrat für später angelegt.

☑ Rad neu erfunden.

☑ Epische Codedokumentation angefertigt.

☑ Früher war alles besser.

☑ Komplexes Code-Meisterwerk erschaffen.

☑ Abhängigkeiten maximiert.

☑ Vorsichtig um Fehler herumgearbeitet.

☑ Wichtige Codestellen ausreichend kopiert.

☑ Kritisches Problem in 5 Minuten gepatcht.

☑ Genügend Codevorrat für später angelegt.

NEIN!

```
/**
 * Executes some very important function.
 */
public void importantMethod() {
    throw new NotImplementedYetException("HAHA!");
}


public void neverEverUsedMethod() {
    //this code is completely untested and
    //might or might not work.
}
```

- Keinen Code auf Vorrat anlegen.
- Ungenutzten / sinnlosen / toten Code löschen (nicht auskommentieren).


☑ Rad neu erfunden.

NEIN!

```
/**
 * Method to encode an Xml-String (replaces & with &amp;
 * @param input any String
 * @return encoded String
 */
public static String encodeXmlString(String input) {
    String output = input;
    output = output.replaceAll("&", "&amp;");
    output = output.replaceAll("\"", "&quot;");
    output = output.replaceAll("<", "&lt;");
    output = output.replaceAll(">", "&gt;");
    output = output.replaceAll("'", "&apos;");
    return output;
}
```

- Basisbibliotheken seiner Plattform kennen
  - Java: java.lang, java.util, jakarta commons, jdom, joda time ...
  - C++: STL, boost...
- Suchen ist langfristig besser als neuschreiben
- Höhere Qualitätsanforderungen als Fachlicher Code

☑ Epische Codedokumentation angefertigt.

NEIN!

```
/**
 * Return the value of the field ID.
 * @return the value of the requested field
 */
public short getId() {
    return id;
}    // method getId
```

- Keine Dokumentation ist besser als schlechte.
- Selbstdokumentierender Code
  - Nein, das ist kein Mythos
  - für sich sprechende Bezeichner
  - in kleinere Stücke aufteilen
- Immer wenn man recherchieren musste: gleich mit dokumentieren.


1
0
☑ Früher war alles besser.

NEIN!

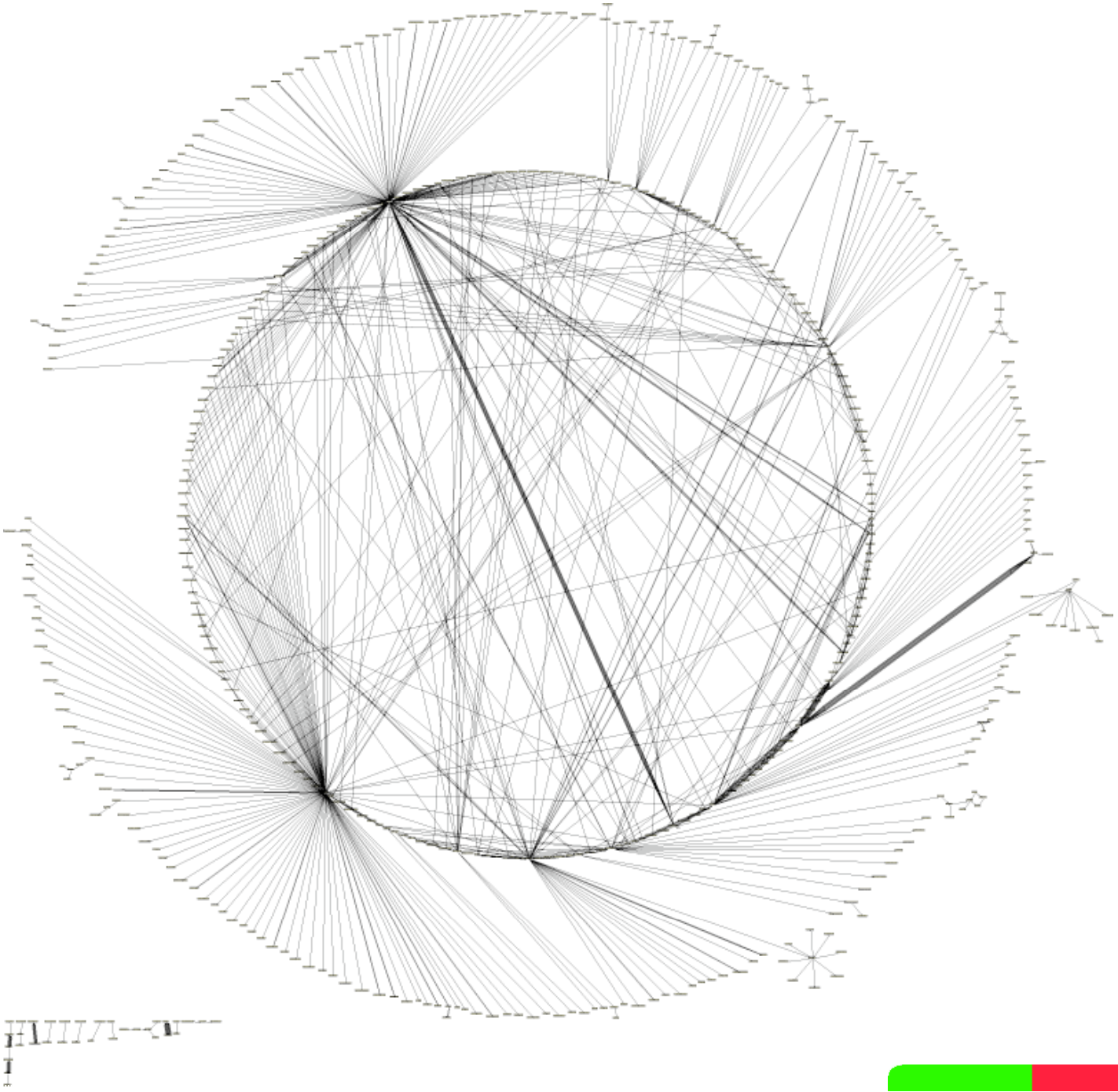
```
#define NO_ERROR 0
#define ERROR_GENERAL 1
#define ERROR_NOT_IMPLEMENTED 2
#define ERROR_SOCK_RESET 3
```

- Mehrere Paradigmen / Sprachen / Methodiken lernen
  - Java, Python / Ruby / JavaScript, C / C++...
  - OOP, AOP, prozedural, deklarativ
  - statische / dynamische Sprachen
  - Agile Softwareentwicklung (XP, Scrum, TDD)
- Ein Buch pro Jahr (mindestens!)

☑ Komplexes Code-Meisterwerk erschaffen.

- Wenn es Dir zu kompliziert vorkommt, ist es das meistens auch.
- „Make everything as simple as possible, but not simpler“
- Metrik-Tools können helfen, allerdings nur als Indikator.

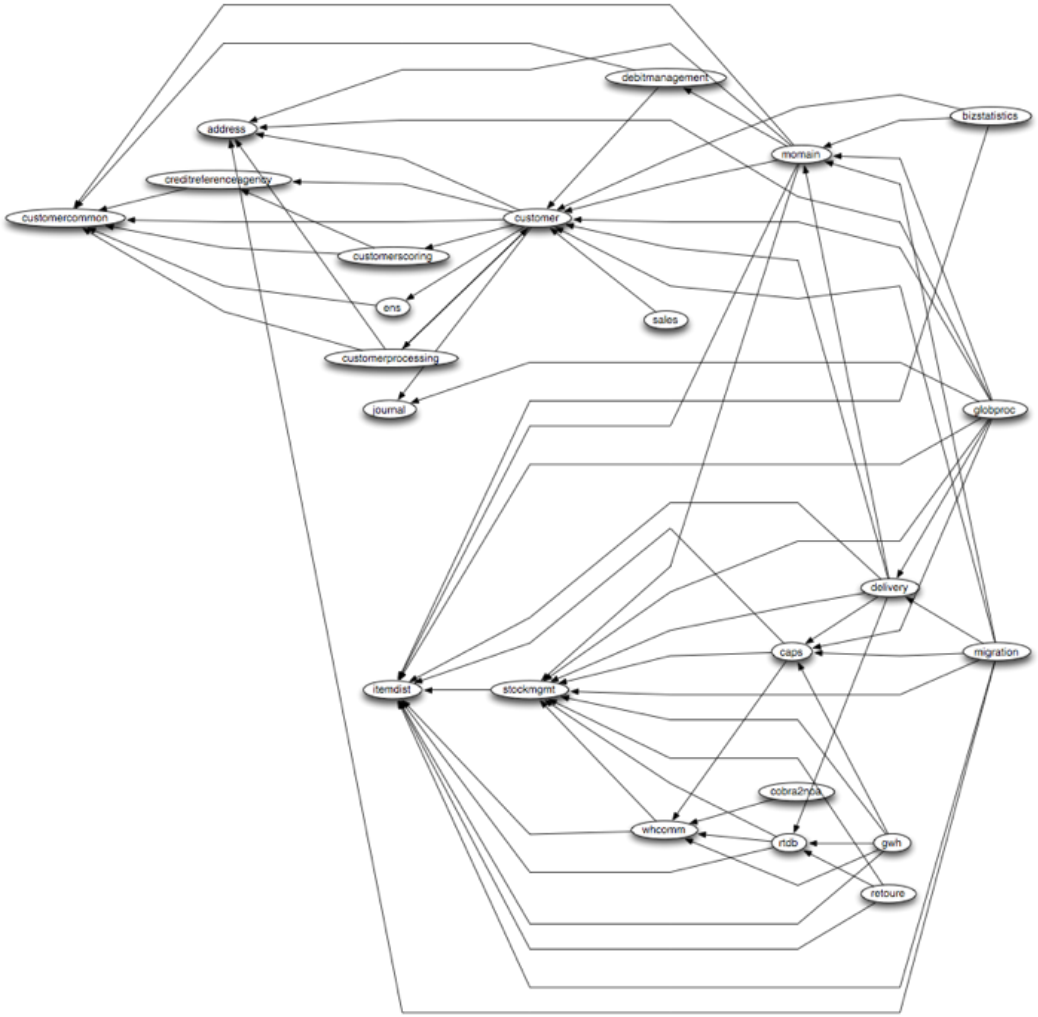

☑ Abhängigkeiten maximiert.


NEIN!

NEIN!


onlineorder
onlinestockmgmt
esebecustomer
rt2noa
onlinecustomer
onlinesyspara
online
standard_saf_wsstub
debitmanagement
clsonline
rtonlinestockmgmt
ens
onlinejournal
customercommon
journal
rtonlinesyspara
esebesyspara
infosys
jess
rtonlinesyspara_wsstub
ens_wsstub_ext
ibatis_dao_2
ibatis_sqlmap_2
ibatis_common_2
cglib_full_2_0_2
onlinejournal_wsstub
ens_wsstub
gwh_wsstub
itemdist_wsstub
online_wsstub
onlinende_wsstub
onlineorder_wsstub
onlinestockmgmt_wsstub
rtdb_wsstub
rtonlinestockmgmt_wsstub
sysspara_wsstub
whcomm_wsstub
wlclient
caps_wsstub
delivery_wsstub
onlinecustomer_
weblogic

Vorhergehender Graph auf „fachliche“ Projekte reduziert



- Anzeichen für Architekturprobleme sind
  - Von einem komplexen Modul werden immer nur Teile benötigt
  - #include "all.h"
  - Verschiedene Versionen der gleichen Bibliothek vorhanden

☑ Vorsichtig um Fehler herumgearbeitet.

- Wenn Du‘s nicht kannst, tu‘s nicht.
- www.google.de
- Nicht rumbasteln.


☑ Wichtige Codestellen ausreichend kopiert.

NEIN!

```
$ grep -R paralelizable * | wc -l
  106
$
```

- Copy and Paste sind böse, mkay?
- Lerne die Refactoring-Features Deiner IDE
- DRY: Don‘t repeat yourself.


☑ Kritisches Problem in 5 Minuten gepatcht.

NEIN!

```
package java.lang;

public class Integer extends Number implements Comparable {

    public Integer addAndMultiply(int add, int mult) {
        ...
    }

}
```

Bildnachweise

- www.photocase.de: „lernen, lernen, pompernen“ (92161, luxuz)
- www.photocase.de: „Stoßdämpfer“ (74034, careaux mit o.)
- www.photocase.de: „Bitte Wenden!“ (86141, NaPra)
- www.photocase.de: „Es werde Licht“ (91748, Punkti)
- www.photocase.de: „Mechanik“ (42402, BrandtMarke)
- www.photocase.de: „Hommage an Miro“ (76393, Frollein S.)
- www.photocase.de: „Nicht anfassen!“ (93900, Lucas88)
- www.photocase.de: „sogar die Wolken scheinen sich zu wiederholen...“ (73068, änte)
- www.photocase.de: „aufgenäht“ (66660, Pippilotta)