Reinventing the Wheel

Patrick Ronecker,
Fabian Danner

Gliederung

1. Allgemeines
2. Ursachen
3. Symptome
4. Konsequenzen
5. Lösungen und Vermeidung
6. Beispiele
7. Zusammenfassung

Allgemein

- Software & System Antipattern
- Implementieren von bereits existierenden Funktionen
- erhebliches Einsparungspotential

Ursachen

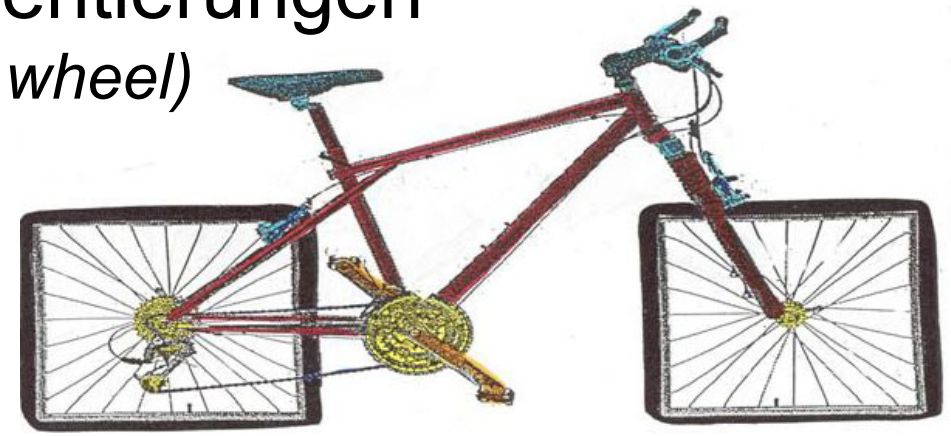
- unerfahrene Programmierer
- Bezahlung nach Lines of Code
- Faulheit beim Recherchieren
- Stolz des Programmieres
- Vermeiden von Lizenzproblemen
- keine Kommunikation zwischen verschiedenen Software Projekten

Symptome

- geschlossene Systemarchitekturen
- unausgereifte und instabile Architekturen
- Nachbildung von exisiterenden Softwarefunktionen
- selbe / ähnliche Funktionsnamen wie in bekannten Bibliotheken

Konsequenzen

- schlechtere Implementierungen
  (Reinventing the square wheel)



- höhere Entwicklungszeiten & Kosten
  - schlechtes Risiko & Zeitmanagement
  - keine Auslieferung der geforderten Funktionen

Lösung und Vermeidung

- Bibliotheken benutzen
- vor implementierung recherchieren

Architecture Mining

- Repräsentative Technologien sammeln und modellieren
- Generalisierung zu einem einheitlichen Design
- Verfeinern des Designs

Beispiele - web.config

Existing	ConfigurationManager.AppSettings["key"]
Reinvented	

```
public static string GetWebConfigValueByKey(string key)
{
    // Load web.config
    XPathDocument xpDoc =
        new XPathDocument(HttpContext.Current.Server.MapPath("~/web.config"));

    // Set up navigator
    XPathNavigator xpNav = xpDoc.CreateNavigator();
    string xpQuery = "//configuration/appSettings/add";
    XPathNodeIterator xpni = xpNav.Select(xpQuery);

    string value = string.Empty;

    // Find the key
    while (xpni.MoveNext())
    {
        if (xpni.Current.GetAttribute("key", xpni.Current.NamespaceURI) == key)
            value = xpni.Current.GetAttribute("value", xpni.Current.NamespaceURI);
    }

    return value;
}
```

Beispiele - bin2hex

Existing

```
int i= Integer.parseInt(bin,2);
hexString=Integer.toHexString(i);
```

Reinvented

```
private static char convertBitsToHexadecimalCharacter(boolean bit1,
                                                      boolean bit2,
                                                      boolean bit3,
                                                      boolean bit4) {

    // if the first bit is true - the binary nibble is 1???
    if (bit1) {

        // if the second bit is true - the binary nibble is 11??
        if (bit2) {

            // if the third bit is true  - the binary nibble is 111?
            if (bit3) {

                // if the fourth bit is true - the binary nibble is 1111
                if (bit4) {

                    // return the 'F' hexidecimal character
                    return HexadecimalConstants.F;

                    // else the fourth bit is false - the binary nibble is 1110
                } else {

                    // return the 'E' hexidecimal character
                    return HexadecimalConstants.E;

                }

                // else the third bit is false - the binary nibble is 110?
            } else {

                // if the fourth bit is true - the binary nibble is 1101
```

Zusammenfassung

- **AntiPattern Name:** Reinventing the Wheel
- **Auch bekannt als:** Design in a Vacuum, Greenfield System
- **Most Frequent Scale:** System
- **Refactored Solution Name:** Architecture Mining
- **Refactored Solution Type:** Process
- **Root Causes:** Stolz, Ignoranz
- **Unbalanced Forces:** Management of Change, Technology Transfer
- **Anecdotal Evidence:** Unser Problem ist einzigartig

Quellen

http://sourcemaking.com/antipatterns/reinvent-the-wheel

http://www.antipatterns.com/arch_cat.htm

http://www.selikoff.net/2009/12/10/why-reinvent-the-wheel/

http://en.wikipedia.org/wiki/Reinventing_the_wheel

http://thedailywtf.com/articles/Right-Under-your-Nose

http://thedailywtf.com/articles/To_the_Hexth_Degree