

2 Refaktorisierung

Any fool can write code that a computer can understand. Good programmers write code that humans can understand. (Refactoring Improving Design of Existing Code)

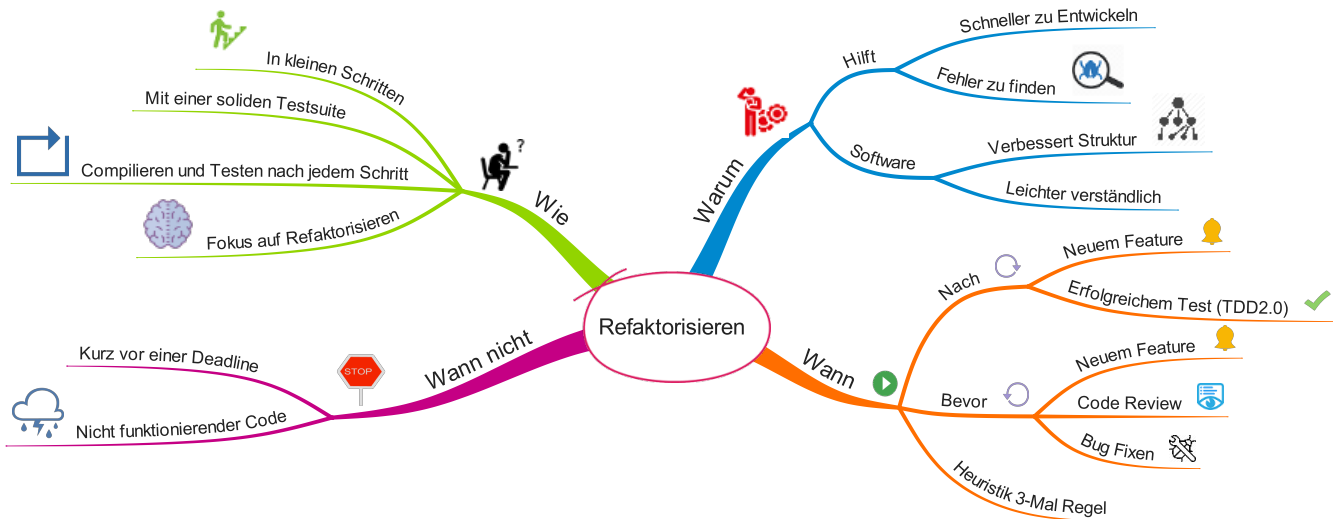
Definition. Eine **Refaktorisierung** ist eine Änderung an der internen Struktur von Software, um das Verständnis zu erleichtern und die Modifizierung zu vereinfachen, ohne das beobachtbare Verhalten zu ändern.

Definition. **Refaktorisieren** beschreibt die Restrukturierung von Software durch Anwenden von einer Reihe von Refaktorisierungen ohne das beobachtbare Verhalten zu ändern.

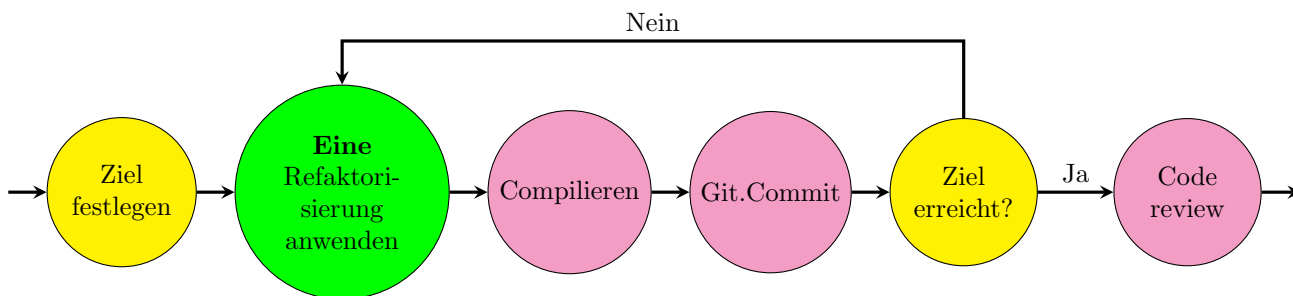
Es geht darum der Degenerierung der Struktur der Software durch geänderte Anforderungen entgegenzuwirken. Es ist Teil des Arbeitsalltags und **nicht** eine spezielle Aufgabe am Ende der Featureentwicklung.

Ziele

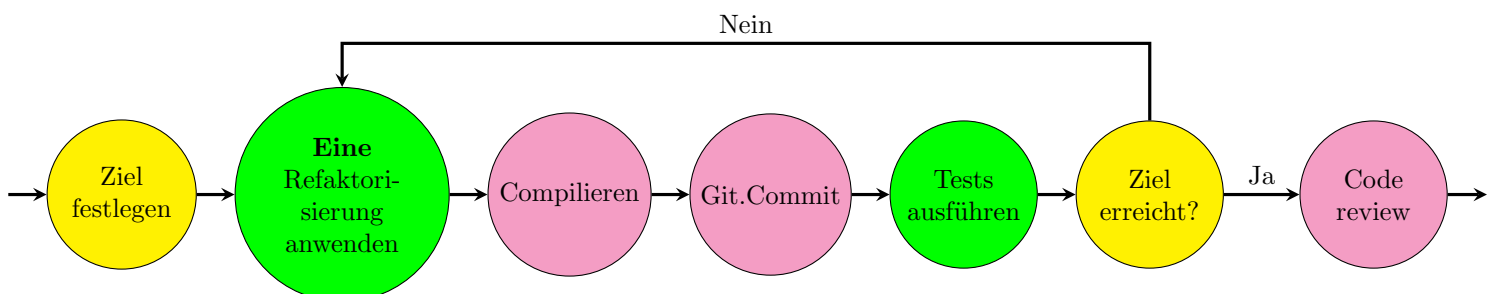
Jedes Refaktorisieren muss ein klar abgegrenztes Ziel haben und zu sinnvollen Zeitpunkten erfolgen.



Ablauf ohne Tests



Ablauf mit Tests



Einfache Refaktorisierungen

vorher	nachher
Umbenennen	
<pre>double proc(Point p1, Point p2, double h) { double b = math.length(p2 - p1); return 0.5 * b * h; }</pre>	<pre>double TriangleArea(Point a, Point b, double height) { double base = math.length(a - b); return 0.5 * base * height; }</pre>
Methode extrahieren	
<pre>if (date.before(SUMMER_START) date.after(SUMMER_END)) { cout << "*****\n"; cout << "** Customer Owes **\n"; cout << "*****\n"; }</pre>	<pre>if (notSummer(date)) { printBanner(); }</pre>
Beschreibende Variable einführen	
<pre>return quantity*itemPrice - max(0, quantity-500)*itemPrice*0.05 + min(quantity*itemPrice*0.1,100.0);</pre>	<pre>double basePrice = quantity*itemPrice; double quantDiscount=max(0,quantity-500)*itemPrice*0.05; double shipping = min(basePrice*0.1,100.0); return basePrice - quantDiscount + shipping;</pre>
Ersetze Daten durch Objekt	
<pre>void rotate(double& x, double& y, double& z, double a);</pre>	<pre>void rotate(Vector3& p, double a);</pre>
Ersetze <i>Magic Number</i> durch symbolische Konstanten	
<pre>return mass * 9.81 * height;</pre>	<pre>const double GRAVITATIONAL_CONSTANT = 9.81; return mass * GRAVITATIONAL_CONSTANT * height;</pre>
Methode verschieben	
<pre>class Calculator { Expression Multiply(Expression, Expression); void Print(Expression); }; class Printer { void Print(int); };</pre>	<pre>class Calculator { Expression Multiply(Expression, Expression); }; class Printer { void Print(Expression); void Print(int); };</pre>
Parameter entfernen	
<pre>void PrintHello(string name, bool b) { cout << "Hello " << name; }</pre>	<pre>void PrintHello(string name) { cout << "Hello " << name; }</pre>
Ersetze Variable durch Aufruf	
<pre>double basePrice = quantity * itemPrice; if (basePrice > 1000) return basePrice * 0.95; else return basePrice * 0.98;</pre>	<pre>if (BasePrice() > 1000) return BasePrice() * 0.95; else return BasePrice() * 0.98; ... double BasePrice() { return quantity * itemPrice; }</pre>