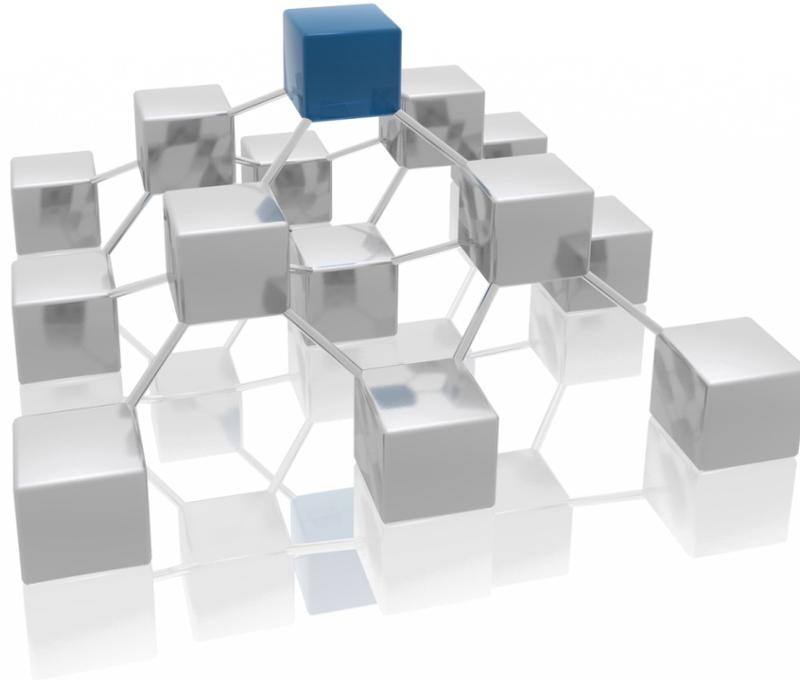


## Softwaremodellierung innerhalb eines SAP ABAP Projekts im agilen Umfeld

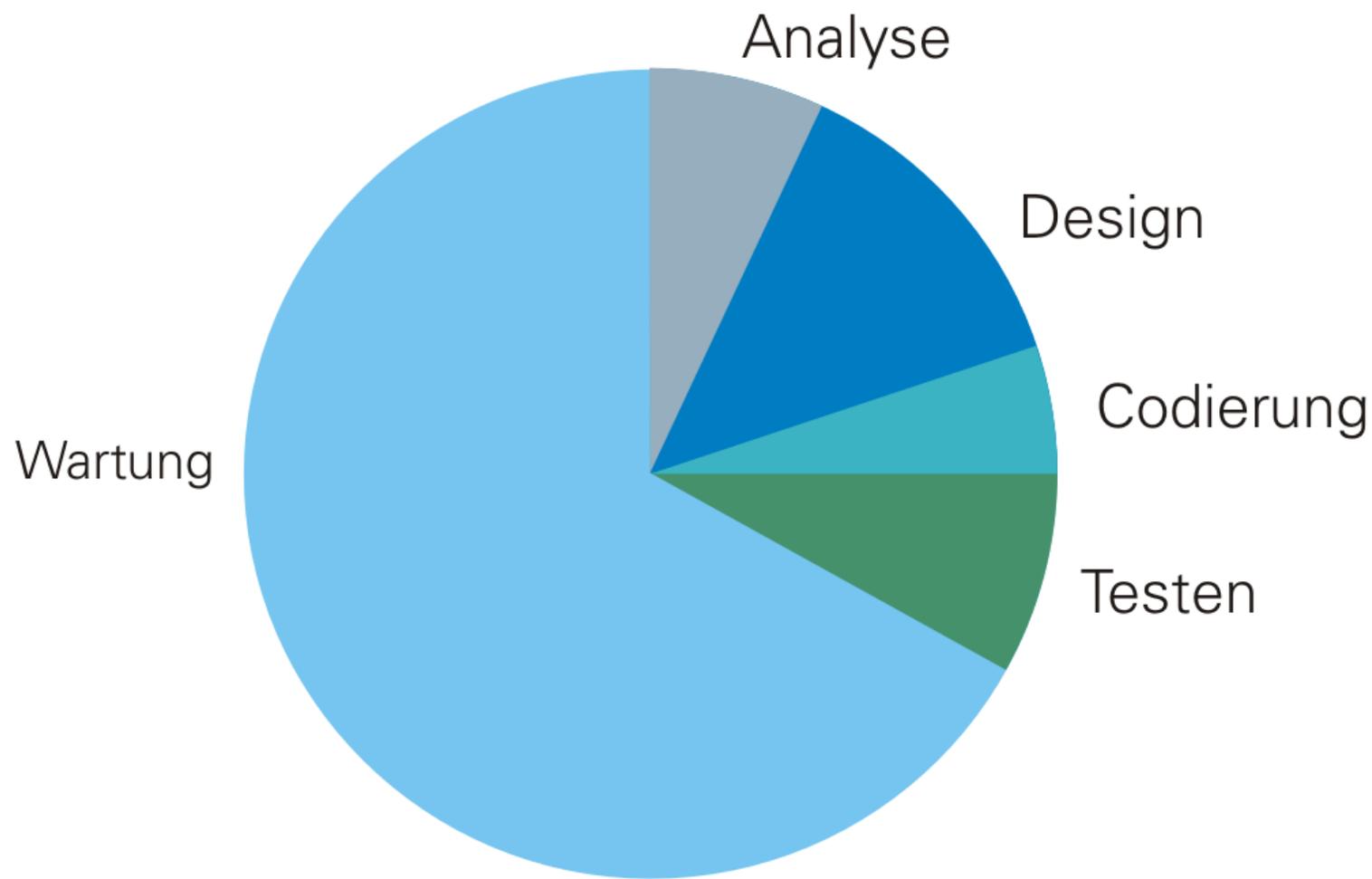


# 1. Die Kosten der Softwareentwicklung

„Warum es manchmal sinnvoll ist, am Anfang mehr zu tun, als nötig ist.“

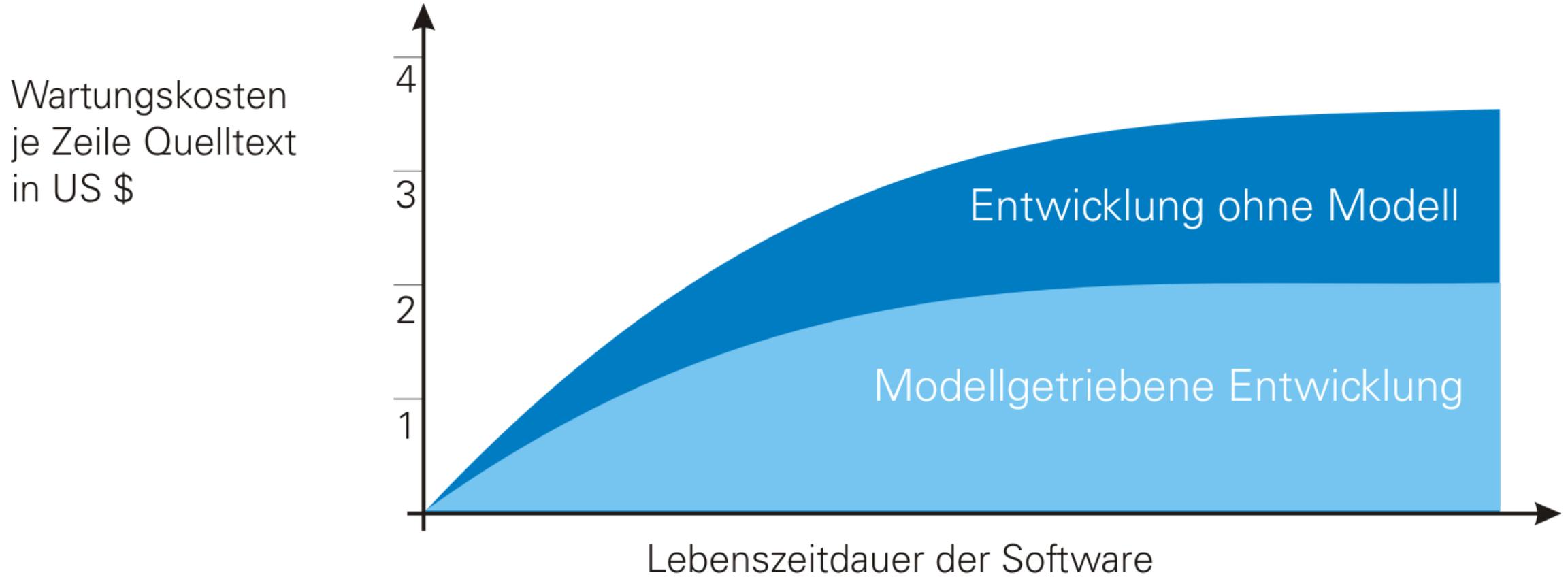
Modellgetriebenen Software-Entwicklung  
im Kostenvergleich

# Kostenarten im Verlauf eines SAP®-Projektes



Nach Tom deMarco „Controlling Software Projects“

# Wartungskosten während der Laufzeit eines SAP®-Projektes



Nach „Crash Report 2011/2012“

- Steigerung der technischen Qualität des Quelltextes
  - Reduzierung des Wartungsaufwandes
    - Übersichtlicher
    - Schneller
    - Reduktion von Seiteneffekten  
(Bugfixing in Methode A führt zu unerwünschten Effekten in Methode B einer anderen Klasse, die ohne UML-Modell schwer aus dem Quelltext ersichtlich sind)

- Steigerung der Akzeptanz der Software beim Anwender, weil:
  - Funktional richtig (textuelle Anforderung i.V.m. Abstraktion durch Modell)
  - Technisch qualitativ hochwertig
    - weniger fehleranfällig
    - Schneller wartbar

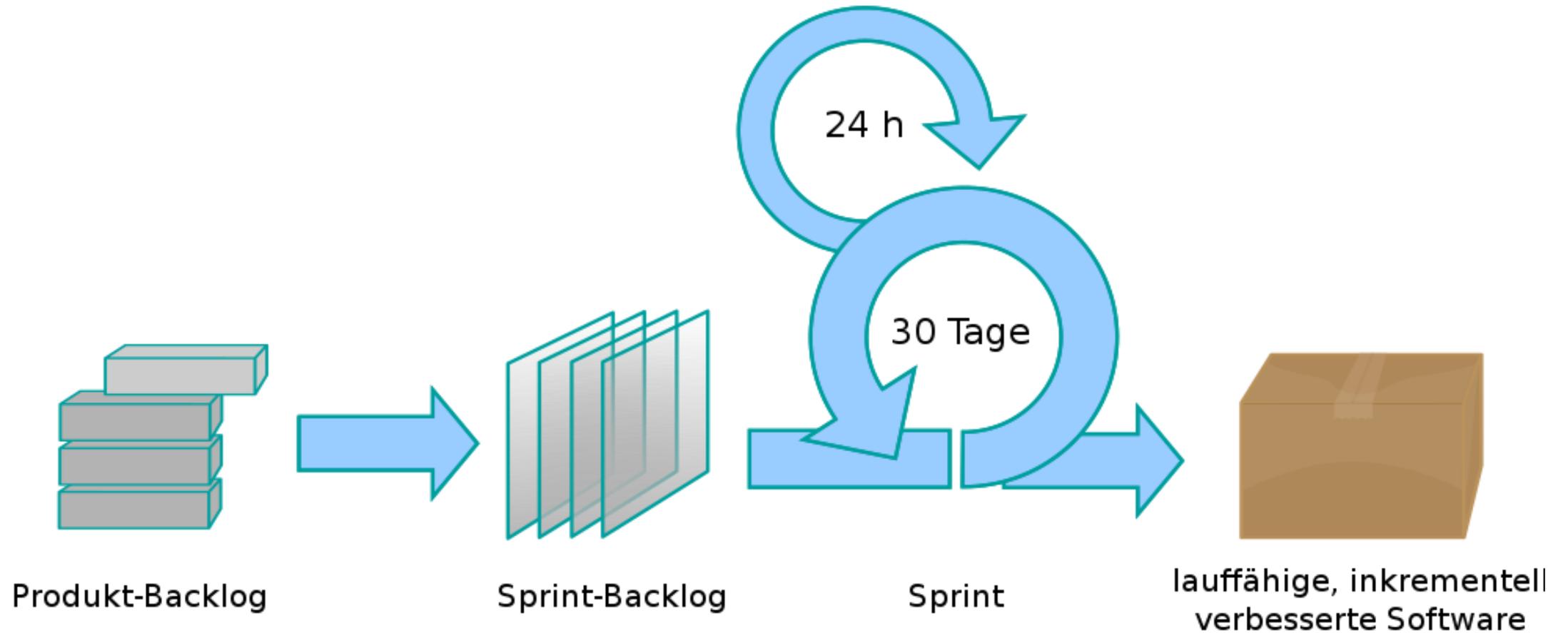
- Reduzierung der Risiken der funktionalen Fehlentwicklung
  - Verbesserte Kommunikation zwischen Anwender und Techniker auf Basis der Abstraktionsebene in einer einheitlichen Sprache (UML)

## 2. Der „agile“ Ansatz

Ziel:

- Schnelle Iterationszyklen
- Dynamische Entwicklung
- Laufende Qualitätsprüfungen

# Der Scrum-Prozess:



„Scrum akzeptiert, dass der Entwicklungsprozess nicht vorherzusehen ist.

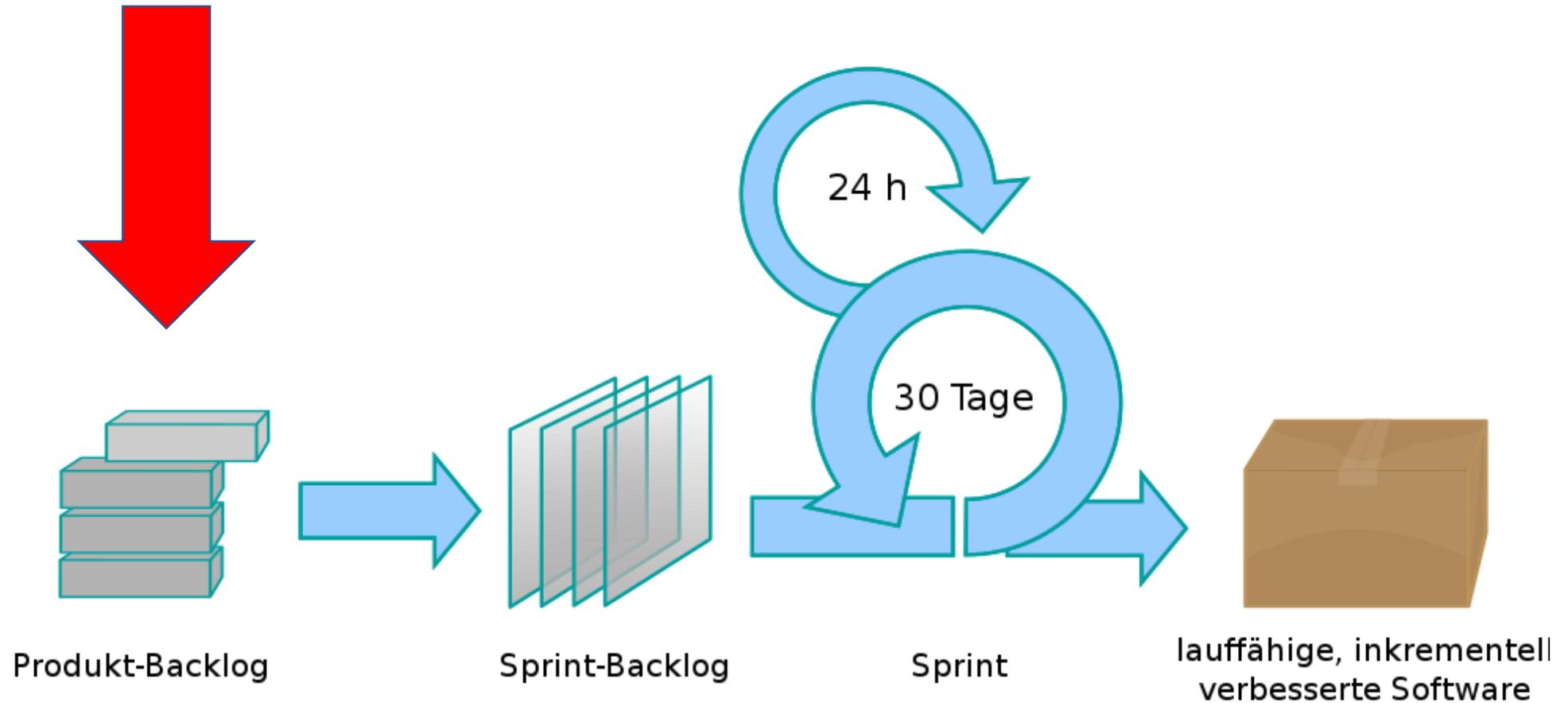
Das Produkt ist die bestmögliche Software unter Berücksichtigung der Kosten, der Funktionalität, der Zeit und der Qualität.“

Ken Schwaber

3. Wie gehen „Scrum“ und „Modellgetriebene Softwareentwicklung“ zusammen?

Kann auch unter dem „Dinosaurier“ ABAP erfolgreich agil und modelliert entwickelt werden?

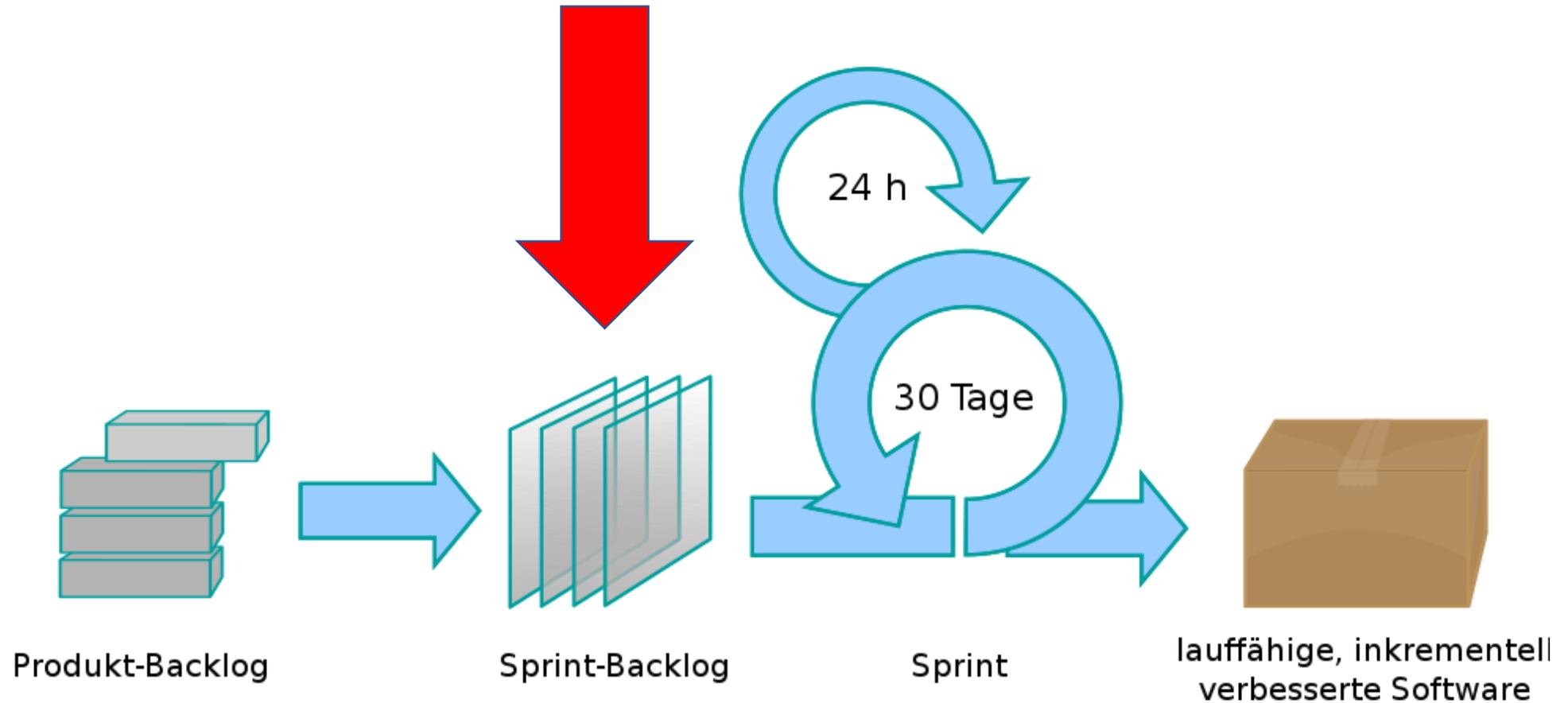
# Produkt-Backlog



# Produkt-Backlog

- Zuständigkeit: Product Owner
- Enthält die Anforderungen an das Projekt
- Neue Anforderungen können jederzeit aufgenommen werden
- Alter Anforderungen können jederzeit gestrichen werden
- Enthält Priorisierung der Anforderungen
- Enthält Aufwandsschätzung zu jeder Anforderung

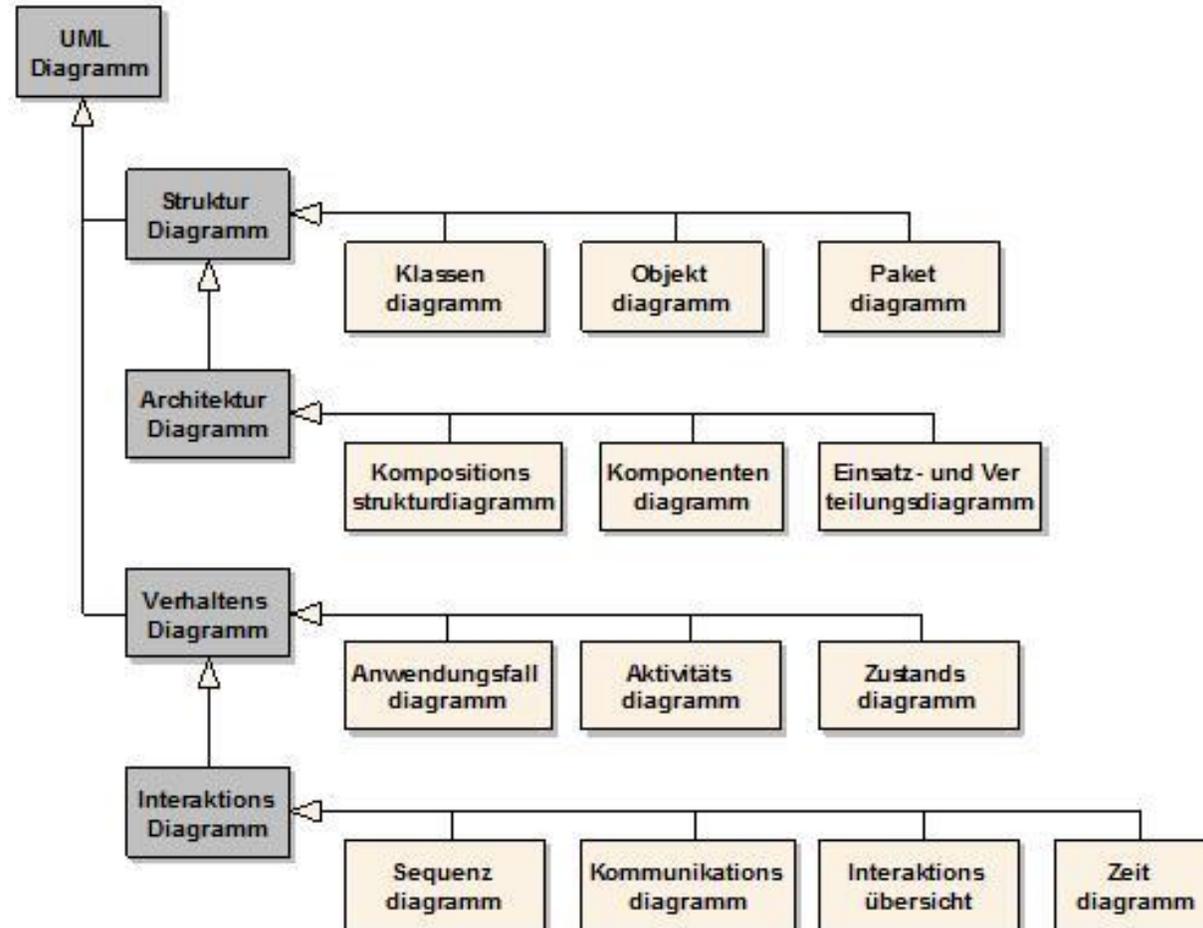
# Sprint-Backlog



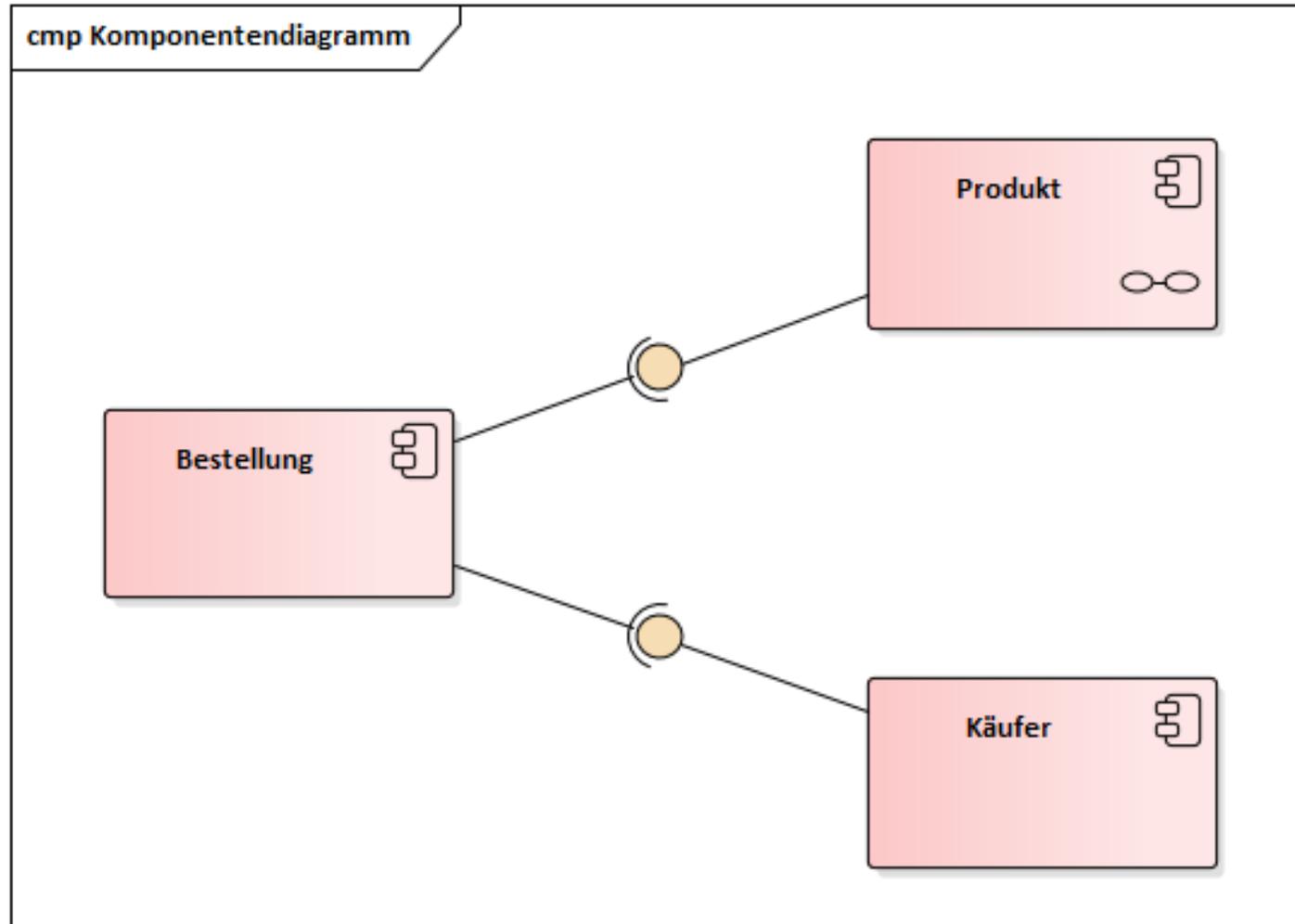
# Sprint-Backlog

- Zuständigkeit: Scrum Master
- Enthält die Anforderungen des aktuellen Sprints, heruntergebrochen auf Tasks (Arbeitspakete)
- Verschiebung der Anforderungen zurück in den Produkt-Backlog möglich, sofern Sprint gefährdet
- Anforderungen im Sprint Backlog dürfen während des Sprint nicht angepasst werden

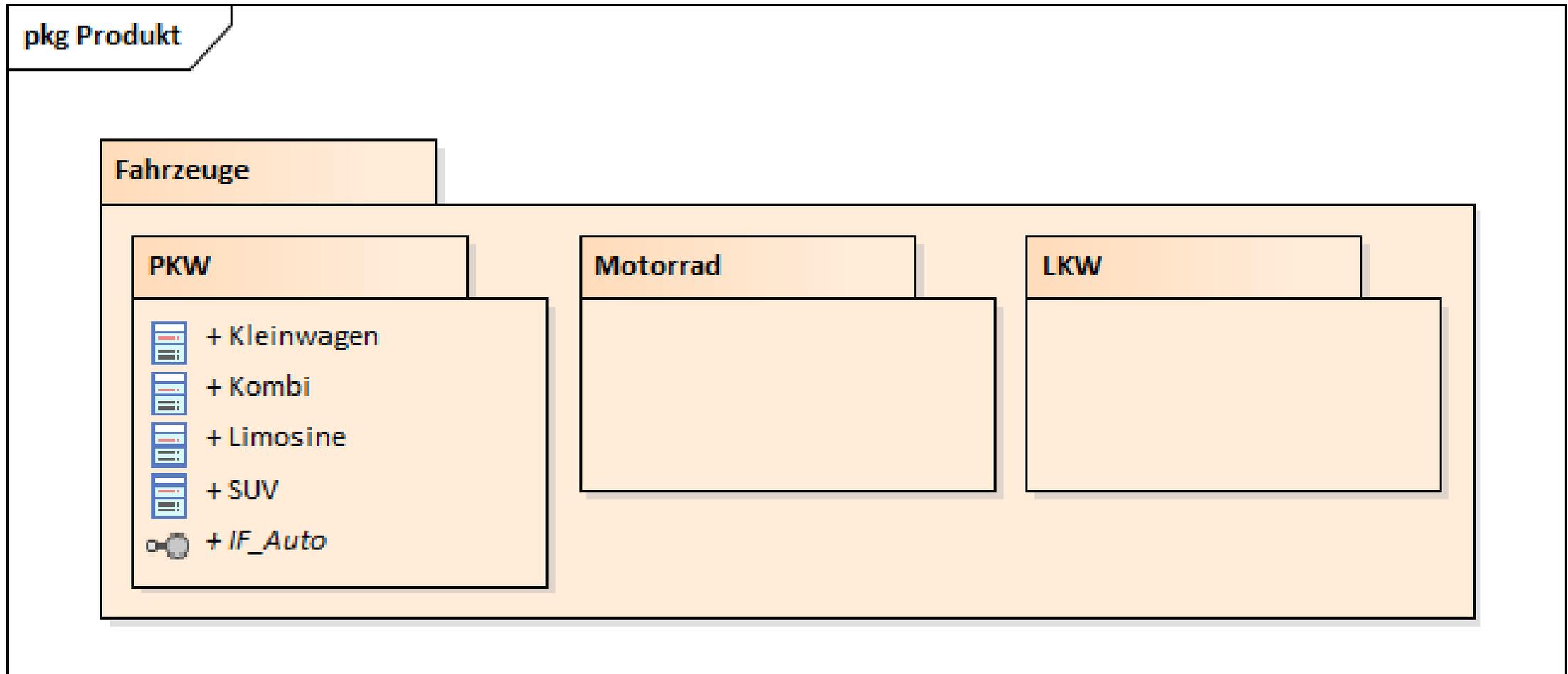
# UML-Diagramme



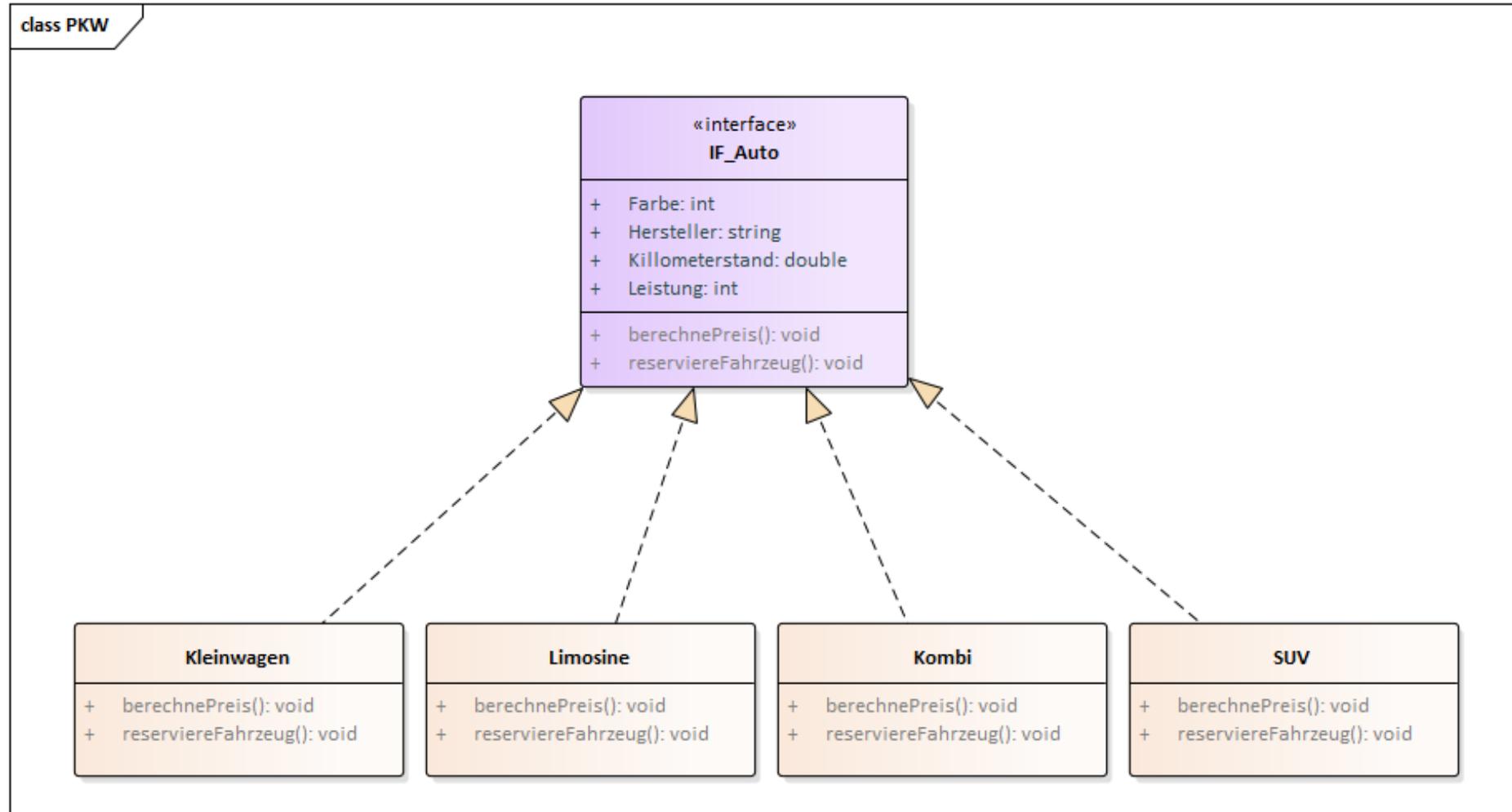
# Komponentendiagramm



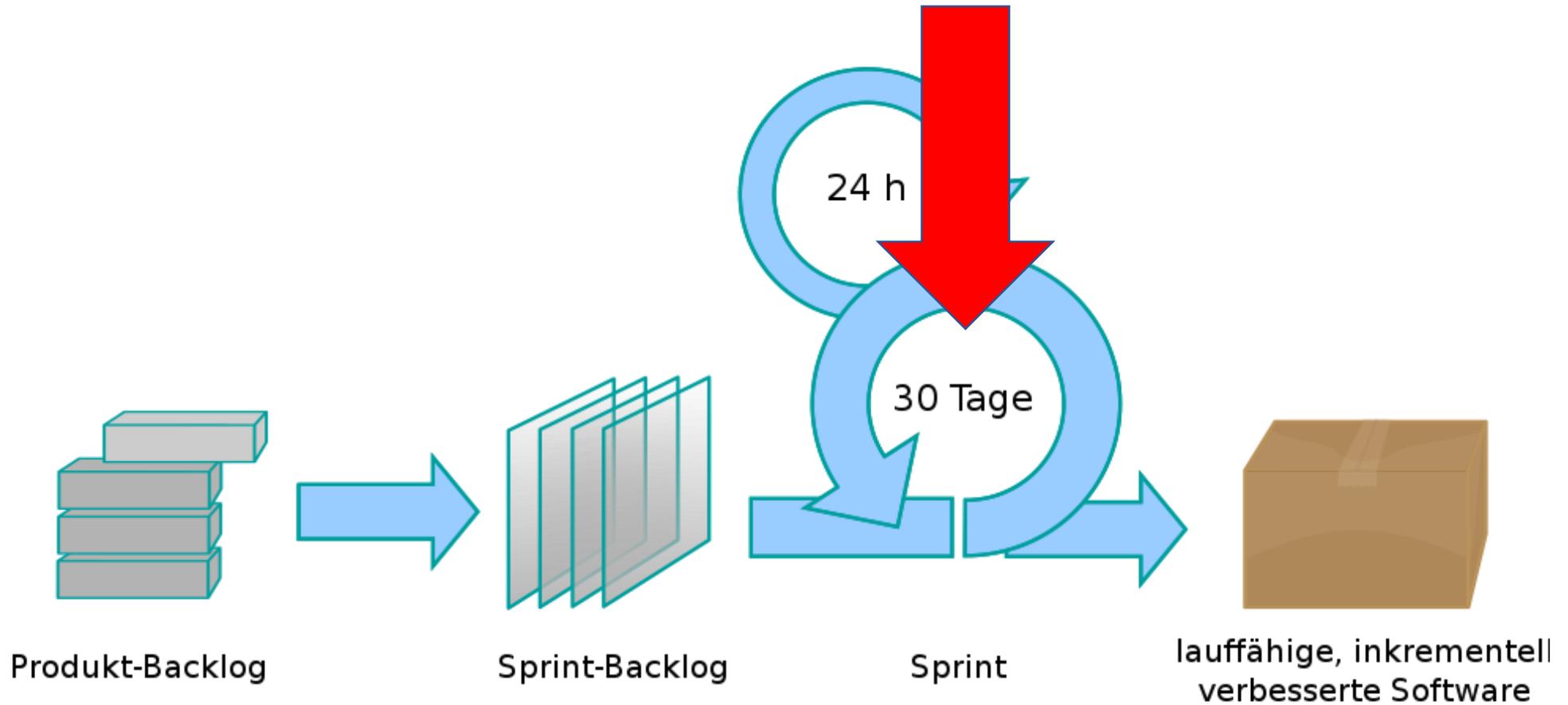
# Paketdiagramm



# Klassendiagramm



# Sprint



Quelle: Wikipedia

# Sprint

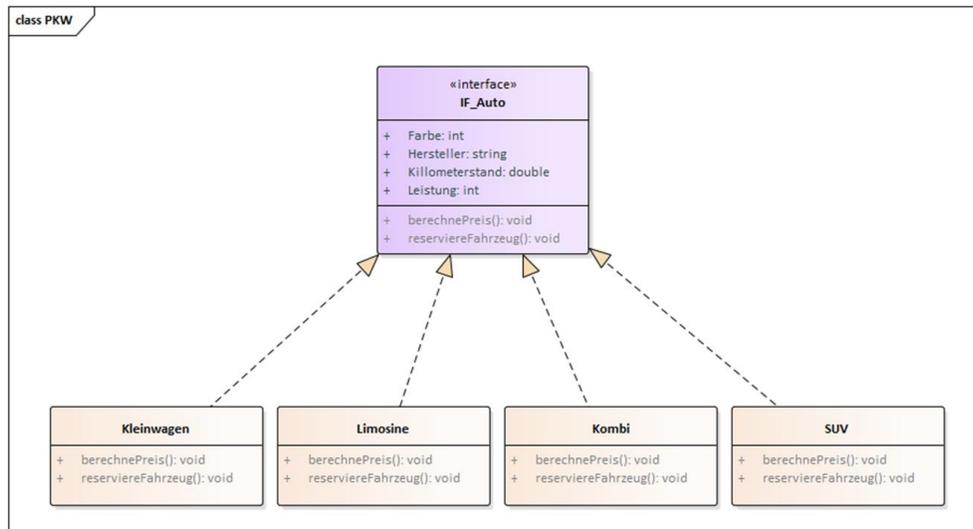
- Zuständigkeit: Entwickler
- Abarbeitung der Tasks
- Regelmäßige Meeting
- Am Ende eines Sprint muss ein funktionsfähiges Zwischenprodukt existieren

# Definition of Done

- Überwacht durch Scrum Master
- Liste von Fertigstellungskriterien
  - Anforderungen / Aspekte
  - Test
  - Dokumentation
  - Überführung des Stands von Entwicklung bis nach Produktion

# Forward Engineering

## Abstrakte Form



## Konkrete Form

```

CL_SALV_TABLE
  CL_SALV_TABLE > FACTORY
  create object lr_cor
  542   exporting
  543     t_choice = lt_choice.
  544
  545   data: lr_salv_content type ref to cl_salv_form_element.
  547   lr_salv_content = me->get_top_of_list( ).
  548   create object lr_form_tol.
  549   lr_form_tol->set_content( lr_salv_content ).
  550
  551   lr_salv_content = me->get_end_of_list( ).
  552   create object lr_form_eol.
  553   lr_form_eol->set_content( lr_salv_content ).
  554
  555   data: ls_hype type if_salv_export=>s_type_hlink.
  556   data: l_column type lvc_fname.
  557   l_column = me->r_columns->GET_HYPERLINK_ENTRY_COLUMN( ).
  558
  559   * ls_hype-t_hype = mt_hyperlinks[ ].
  560   data: lr_hyperlinks type ref to cl_salv_hyperlinks.
  561
  562   ls_hype-hlink_colname = l_column.
  563   lr_hyperlinks = me->r_functional_settings->get_hyperlinks( ).
  564   ls_hype-t_hype = cl_salv_controller_metadata->get_hyperlinks( lr_hyperlinks ).
  565
  566   if cl_salv_bs_a_xml_base=>get_version( ) eq if_salv_bs_xml=>version_25 or
  567     cl_salv_bs_a_xml_base=>get_version( ) eq if_salv_bs_xml=>version_26.
  568
  569   data: r_result_data type ref to cl_salv_ex_result_data_table.
  570
  571   r_result_data = cl_salv_ex_util=>FACTORY_RESULT_DATA_TABLE(
  572     * T_SELECTED_ROWS = lt_lvc_row
  573     * T_SELECTED_COLUMNS = lt_sel_cols
  574     * T_SELECTED_CELLS = lt_sel_cells
  575     R_DATA = me->r_table
  576     S_LAYOUT = ls_layo
  577     T_FIELDATALOG = lt_fcatt
  578     T_SORT = lt_sort
  579     T_FILTER = lt_filt
  580     * T_HYPERLINKS = me->mt_hyperlinks
  581     * S_CURRENT_CELL = ls_cur_cell
  582     * HYPERLINK_ENTRY_COLUMN = ls_hyper_entry
  583     * DROPDOWN_ENTRY_COLUMN = ls_dropdown_entry
  584     * T_DROPDOWN_VALUES = lt_drddn )
  585     r_top_of_list = lr_form_tol
  586     r_end_of_list = lr_form_eol
  587   ).
  588
  
```

# Forward Engineering

- Transformation einer abstrakten Form in eine Konkrete Form
- Erstellt Rümpfe der Objekte
- Schwierigkeiten:
  - Wie konkret darf modelliert werden?
  - Einhaltung von definierten Mustern

# Vorteile von UML-Diagrammen in der SCRUM-Entwicklung

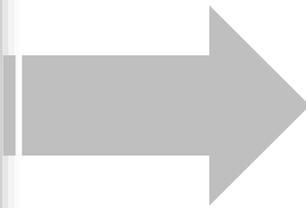
- Strukturierte Übersicht auf das Projekt
- Vereinfachte Ansichten durch Abstraktion
- Zeigen Beziehungen zwischen Elementen
- Einarbeitungszeit neuer Kollegen reduziert
- Abgrenzungen und Beziehungen zwischen Tasks

# Reverse Engineering

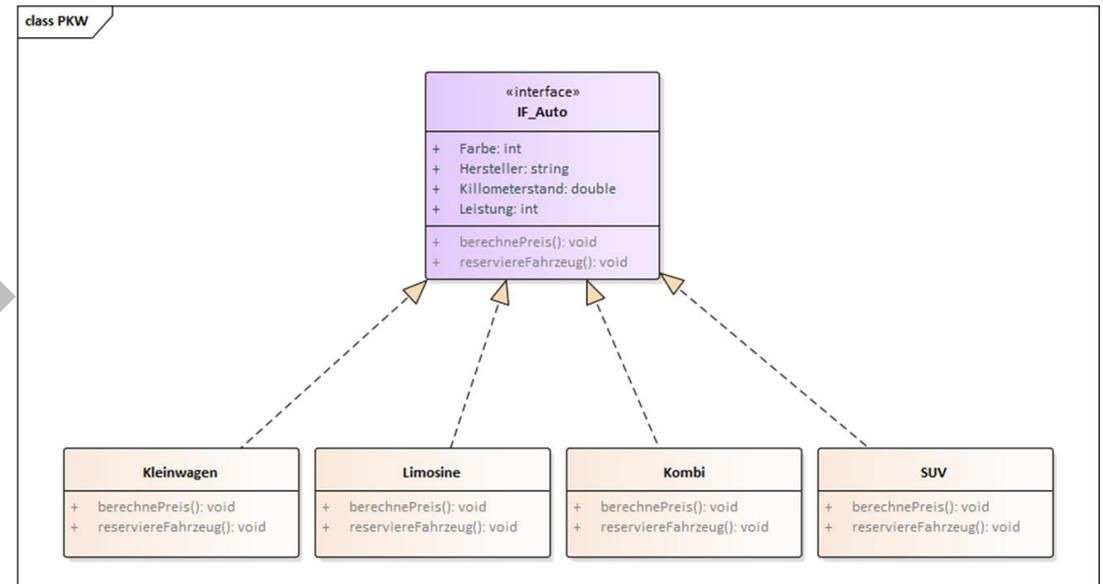
## Konkrete Form

```

] CL_SALV_TABLE
CL_SALV_TABLE > FACTORY
541 create object lr_cor
542     exporting
543         t_choice = lt_choice.
544
545 data: lr_salv_content type ref to cl_salv_form_element.
546
547 lr_salv_content = me->get_top_of_List( ).
548 create object lr_form_tol.
549 lr_form_tol->set_content( lr_salv_content ).
550
551 lr_salv_content = me->get_end_of_List( ).
552 create object lr_form_eol.
553 lr_form_eol->set_content( lr_salv_content ).
554
555 data: ls_hype type if_salv_export=>s_type_hlink.
556 data: l_column type lvc_fname.
557 l_column = me->r_columns->GET_HYPERLINK_ENTRY_COLUMN( ).
558
559 * ls_hype-t_hype = mt_hyperlinks[ ].
560 data: lr_hyperlinks type ref to cl_salv_hyperlinks.
561
562 ls_hype-hlink_colname = l_column.
563 lr_hyperlinks = me->r_functional_settings->get_hyperlinks( ).
564 ls_hype-t_hype = cl_salv_controller_metadata->get_hyperlinks( lr_hyperlinks ).
565
566 if cl_salv_bs_a_xml_base=>get_version( ) eq if_salv_bs_xml=>version_25 or
567 cl_salv_bs_a_xml_base=>get_version( ) eq if_salv_bs_xml=>version_26.
568
569 data: r_result_data type ref to cl_salv_ex_result_data_table.
570
571 r_result_data = cl_salv_ex_util=>FACTORY_RESULT_DATA_TABLE(
572 *   T_SELECTED_ROWS = lt_lvc_row
573 *   T_SELECTED_COLUMNS = lt_sel_cols
574 *   T_SELECTED_CELLS = lt_sel_cells
575 *   R_DATA = me->r_table
576 *   S_LAYOUT = ls_layo
577 *   T_FIELDATALOG = lt_fcat
578 *   T_SORT = lt_sort
579 *   T_FILTER = lt_filt
580 *   T_HYPERLINKS = me->mt_hyperlinks
581 *   S_CURRENT_CELL = ls_cur_cell
582 *   HYPERLINK_ENTRY_COLUMN = ls_hyper_entry
583 *   DROPDOWN_ENTRY_COLUMN = ls_dropdown_entry
584 *   T_DROPDOWN_VALUES = lt_drdrn )
585 *   r_top_of_list = lr_form_tol
586 *   r_end_of_list = lr_form_eol
587 *   ).
588
  
```



## Abstrakte Form



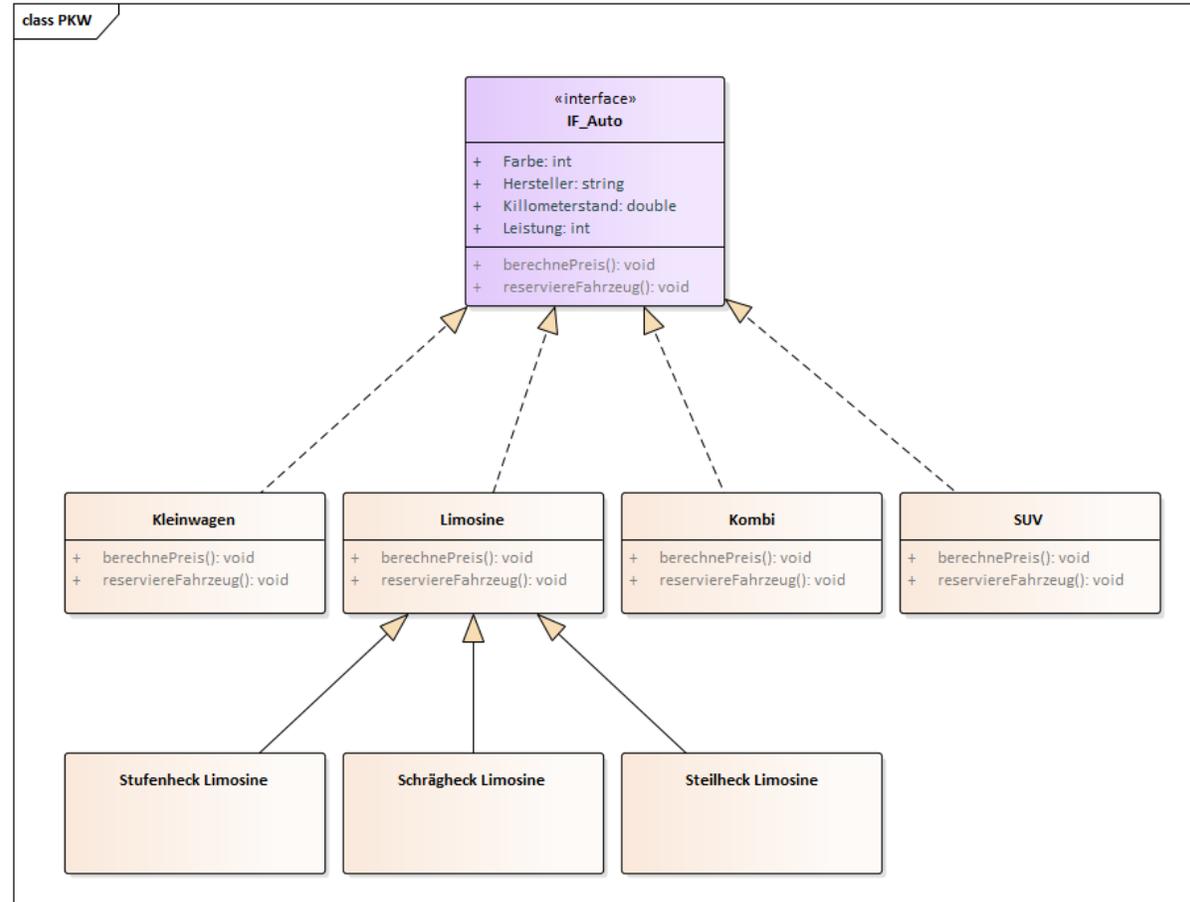
# Reverse Engineering

Reverse Engineering

Startklasse:

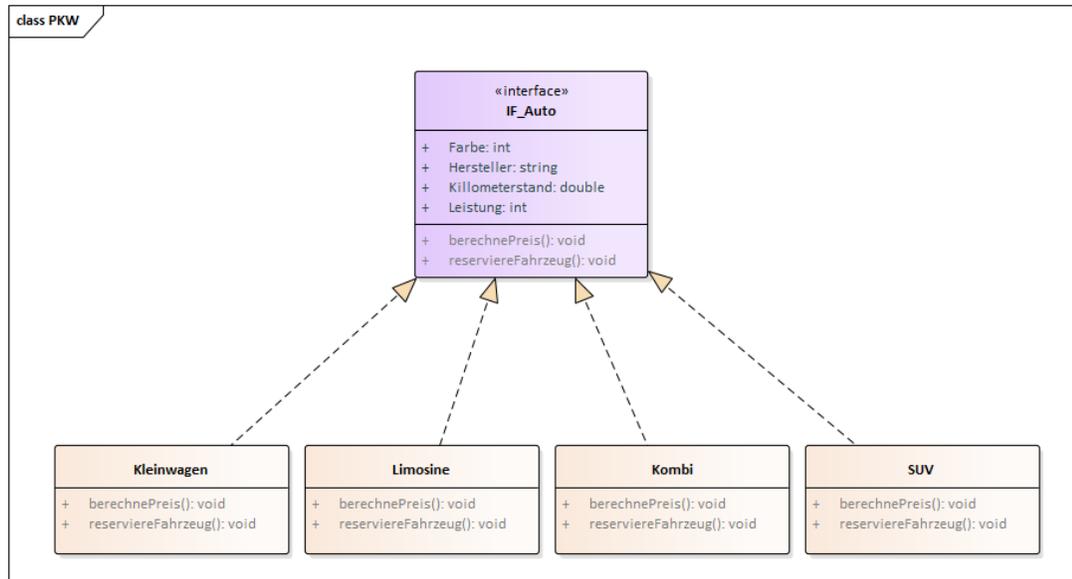
▼

# Reverse Engineering

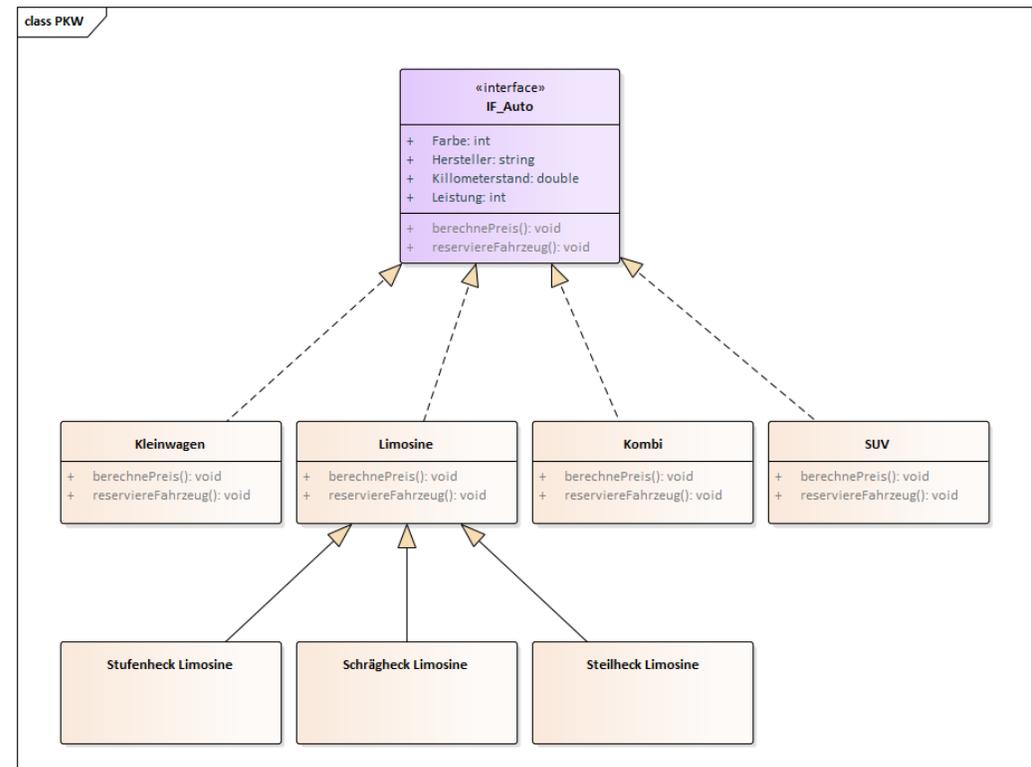


# Reverse Engineering

## Initialmodell



## 1. Abzug



# Baseline

- Versionierung des Projekts zu einem definierten Punkt
- Dient Ermittlung von Unterschieden zwischen zwei Versionen
- Widerspiegelt den Trend der Software

# Baseline

New Baseline

Name: PKW

Version: 1.0  Include sub-packages

Note: Baseline time: 18.09.2018 11:45:19

OK Cancel

Baselines

Package: PKW

Name	Versi	Notes
PKW	1.0	Baseline time: 18.09.2018 11:45:09

Show Differences

Restore to Baseline

New Baseline

Delete Selected

Load Other Baselines ▾

Import File

Export File

Compare Model to File

Options

Close

Help

Project Local - EAP File, ExampleRepository

# Baseline

Baselines

Package: PKW

Name	Versi	Notes
PKW	1.1	Baseline time: 18.09.2018 11:50:45
PKW	1.0	Baseline time: 18.09.2018 11:45:09

Show Differences

Restore to Baseline

New Baseline

Delete Selected

Load Other Baselines ▼

Import File

Export File

Compare Model to File

Options

Close

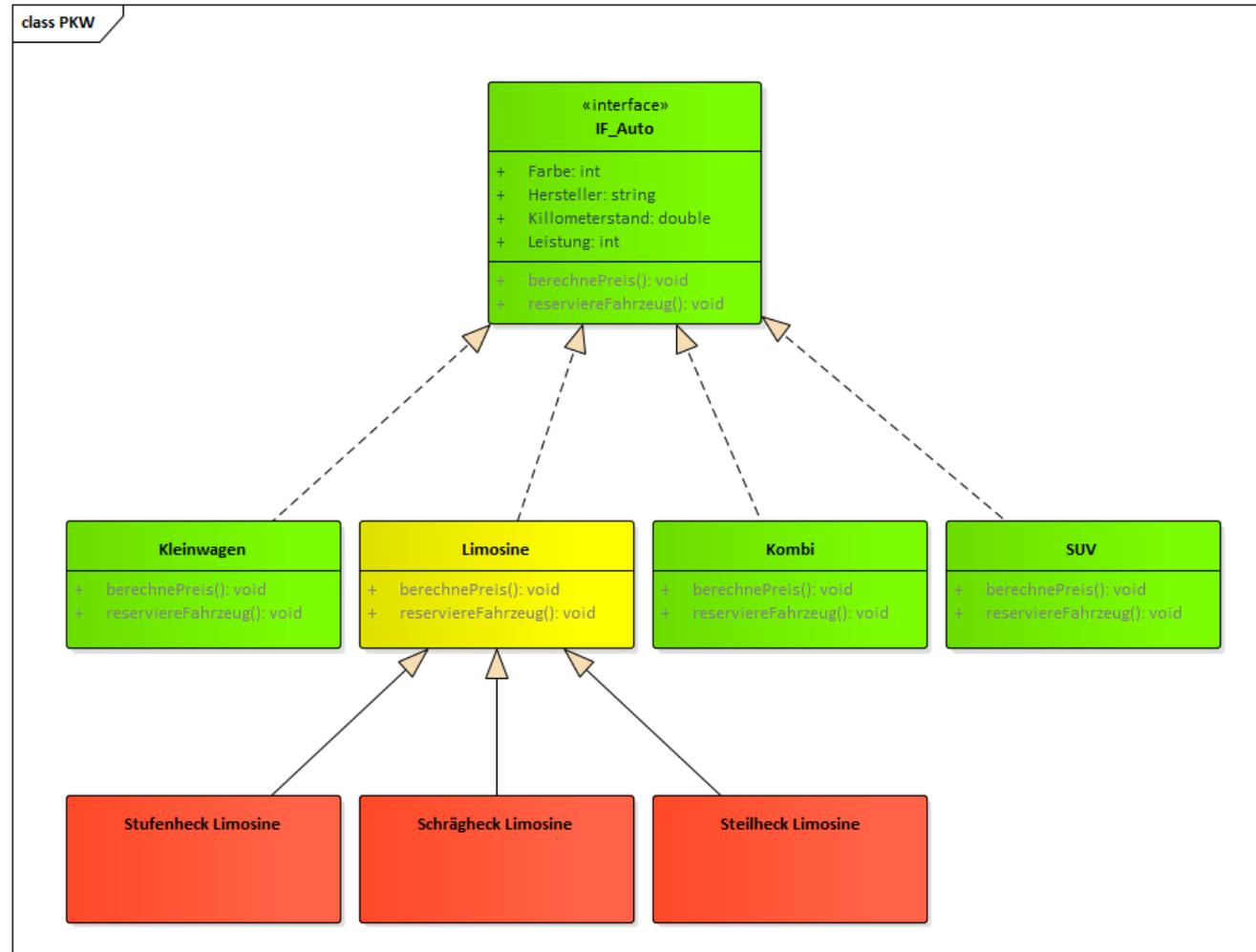
Help

Project Local - EAP File, ExampleRepository

# Baseline

Model Elements	Status	Property	Model	Baseline
PKW		Name	PKW	PKW
Limosine		Type	Logical	Logical
Links		Version	1.0	1.0
Generalization	Model only	Stereotype		
Generalization	Model only	Date Created	14.09.2018 10:53:15	14.09.2018 10:53:15
Generalization	Model only	Date Modified	18.09.2018 11:50:07	15.09.2018 10:34:32
Schrägheck Limosine	Model only	Notes		
Steilheck Limosine	Model only	Style	ShowPrivate=1;ShowProtected=1;Sh...	ShowPrivate=1;Show
Stufenheck Limosine	Model only	StyleEx	ExcludeRTF=0;DocAll=0;HideQuals=0...	ExcludeRTF=0;DocAll-
PKW	Changed		2.1;TExplicitNavigability=0;AdvancedE...	2.1;TExplicitNavigabili

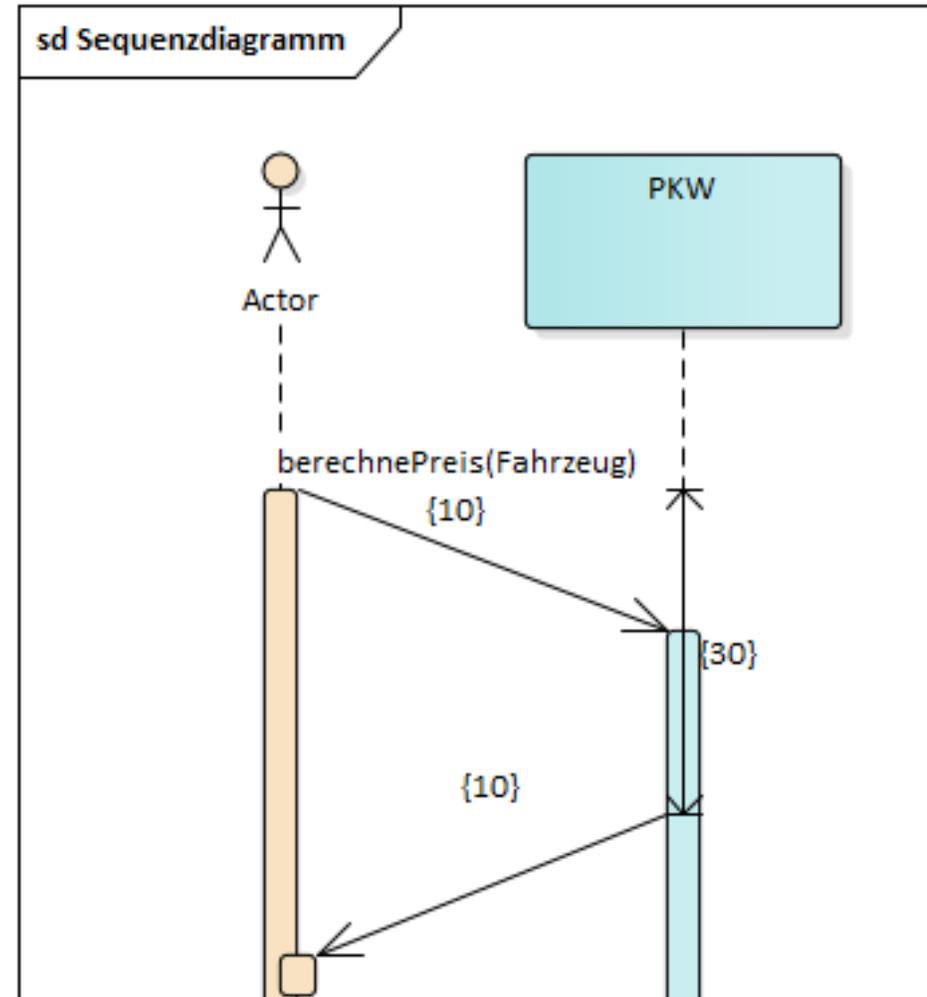
# Baseline



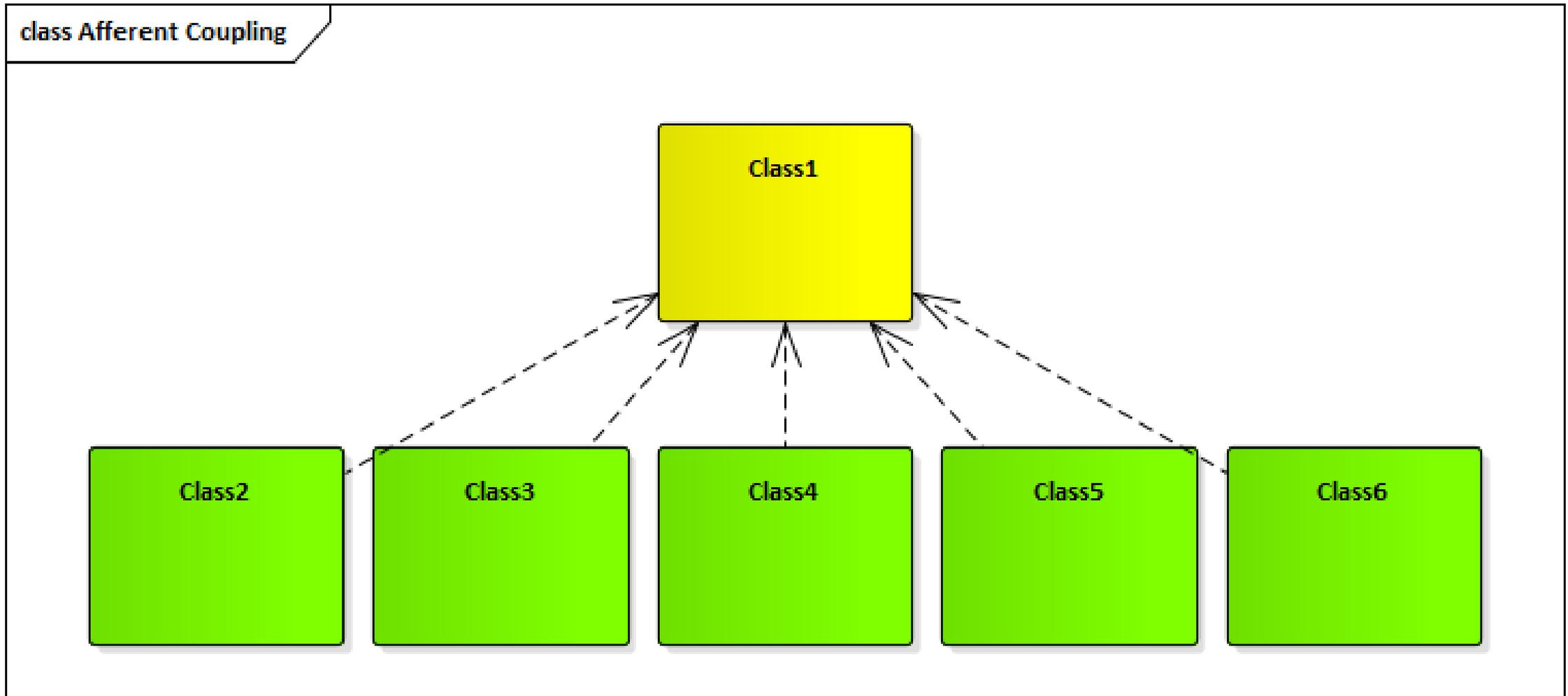
# Analyse durch Metriken

- Bewertung des Quellcodes
- Kontrollmechanismus für die Einhaltung von Anforderungen an das Projekt
- Schwachstellenanalyse
- Softwarequalitätssicherung

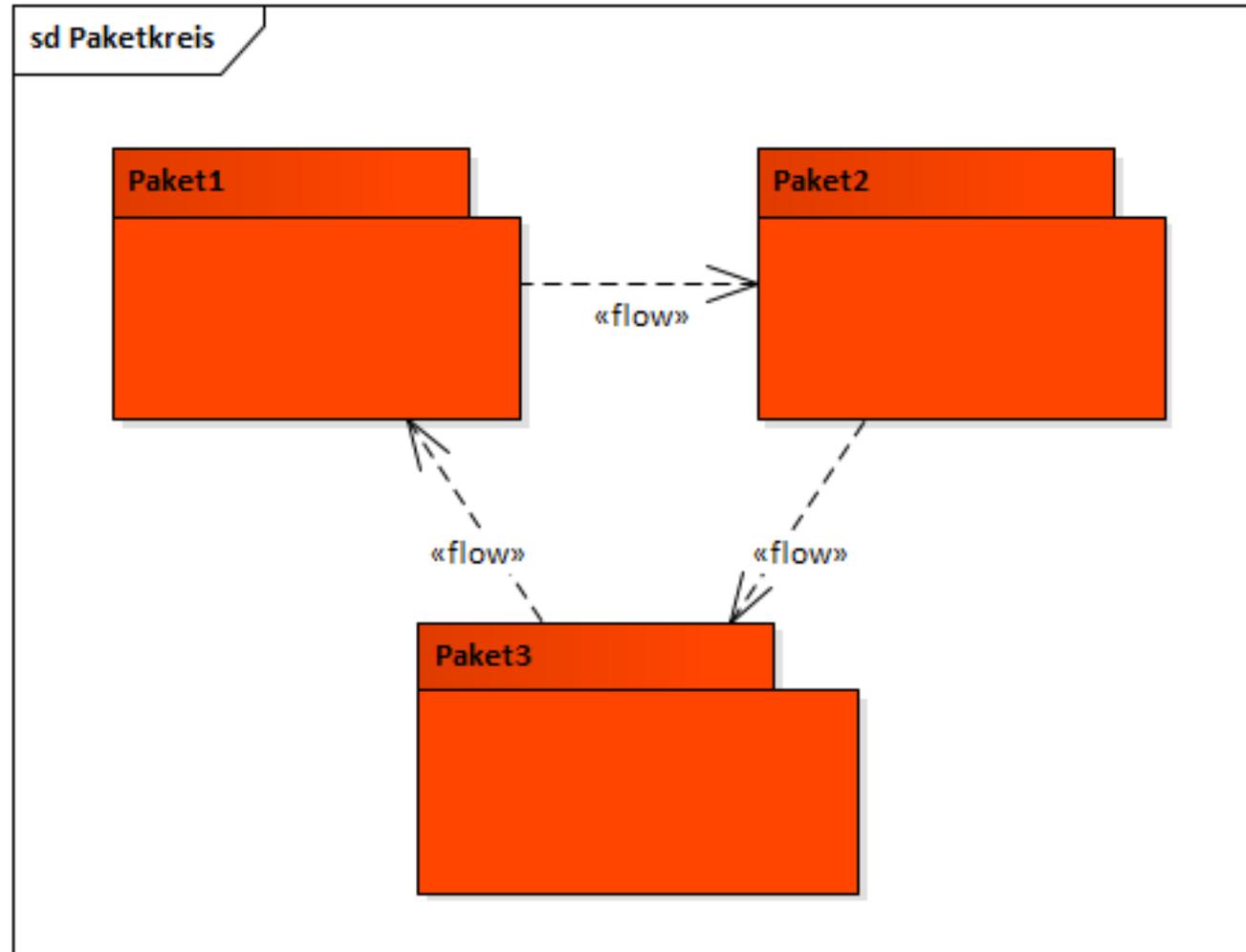
# Performance-Analyse



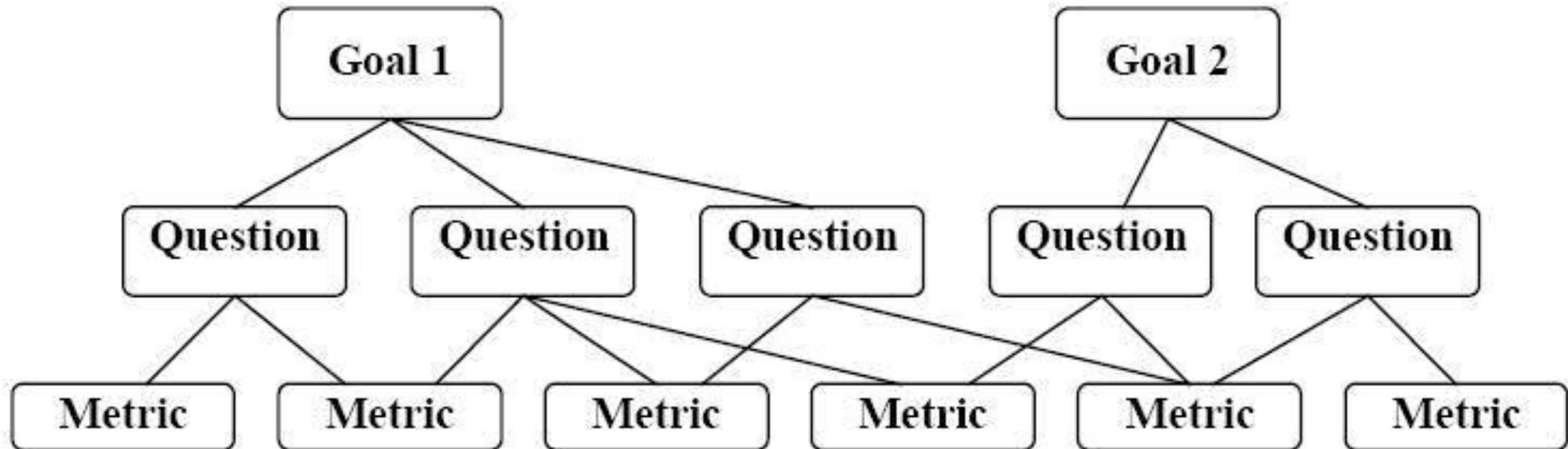
# Afferent Coupling



# Paketkreis



# Goal Question Metric



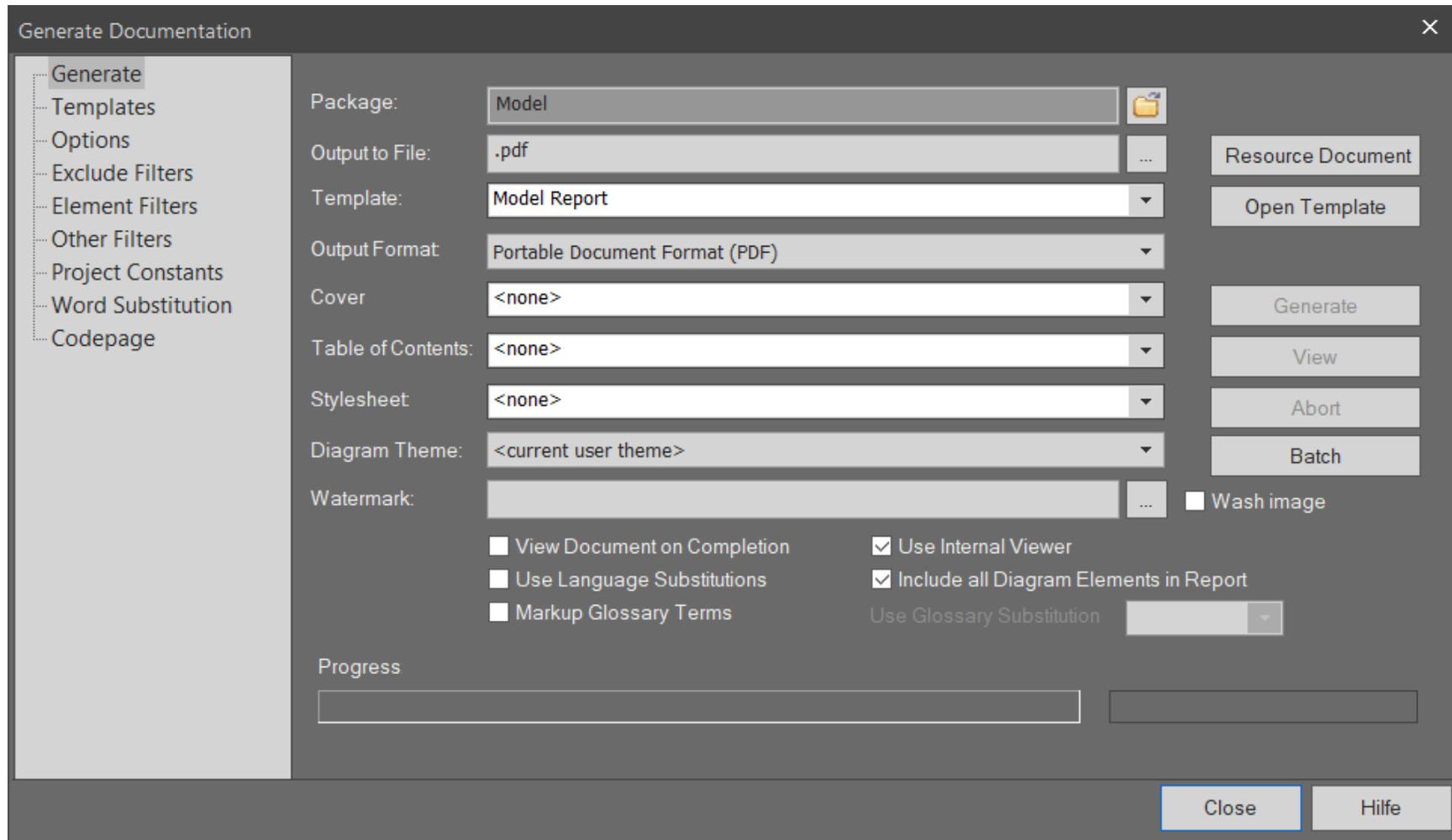
# Goal Question Metric

- Goal: Definiert das zu erreichende Ziel
- Question: Stellt die Frage gegen das Ziel. Unterteilt das Ziel in konkretere Aspekte an das Projekt.
- Metric: Definiert die Metriken, welche die aufgestellten Questions beantworten können.

# „Goal Question Metric“

- G1: Verbesserung der Erreichbarkeit der Webapplikation
  - Q1: Wurde die Reaktionszeit bei Anfragen verbessert?
    - M1: Performance-Analyse anhand der Verarbeitungszeiten
  - Q2: Wurde die Anzahl der benötigten Anfragen reduziert?
    - M2: Anzahl der Requests ermitteln
- G2: Verbesserung der Softwarequalität
  - Q3: Wurde die Wartbarkeit des Projekts verbessert?
    - M3: Afferent Coupling
    - M4: Paketkreis

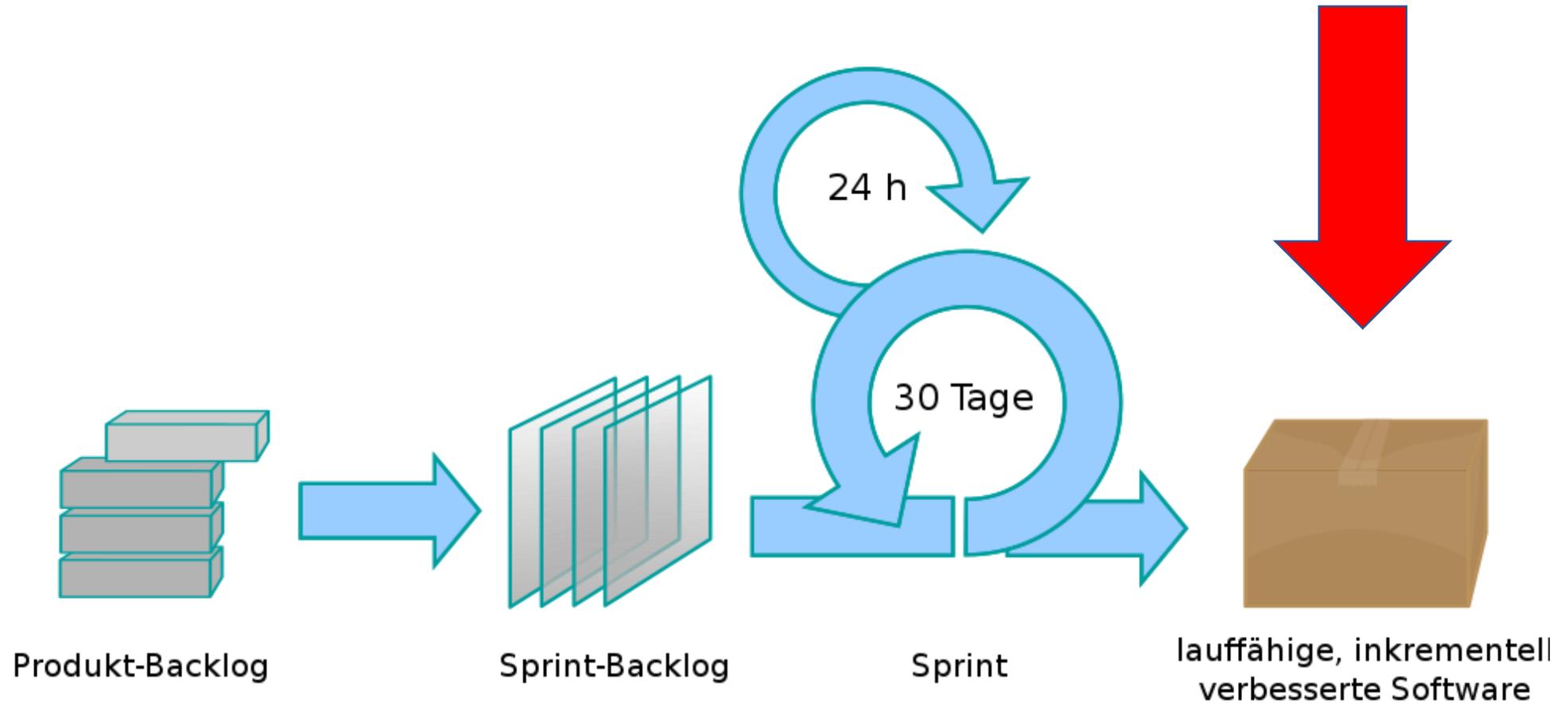
# Dokumentation



# Dokumentation

- Automatisierung zur Erstellung der Dokumentation
- Verwendung von RTF-Templates
- Unterstützte Formate: DOCX, PDF, RTF

# Lauffähige Software

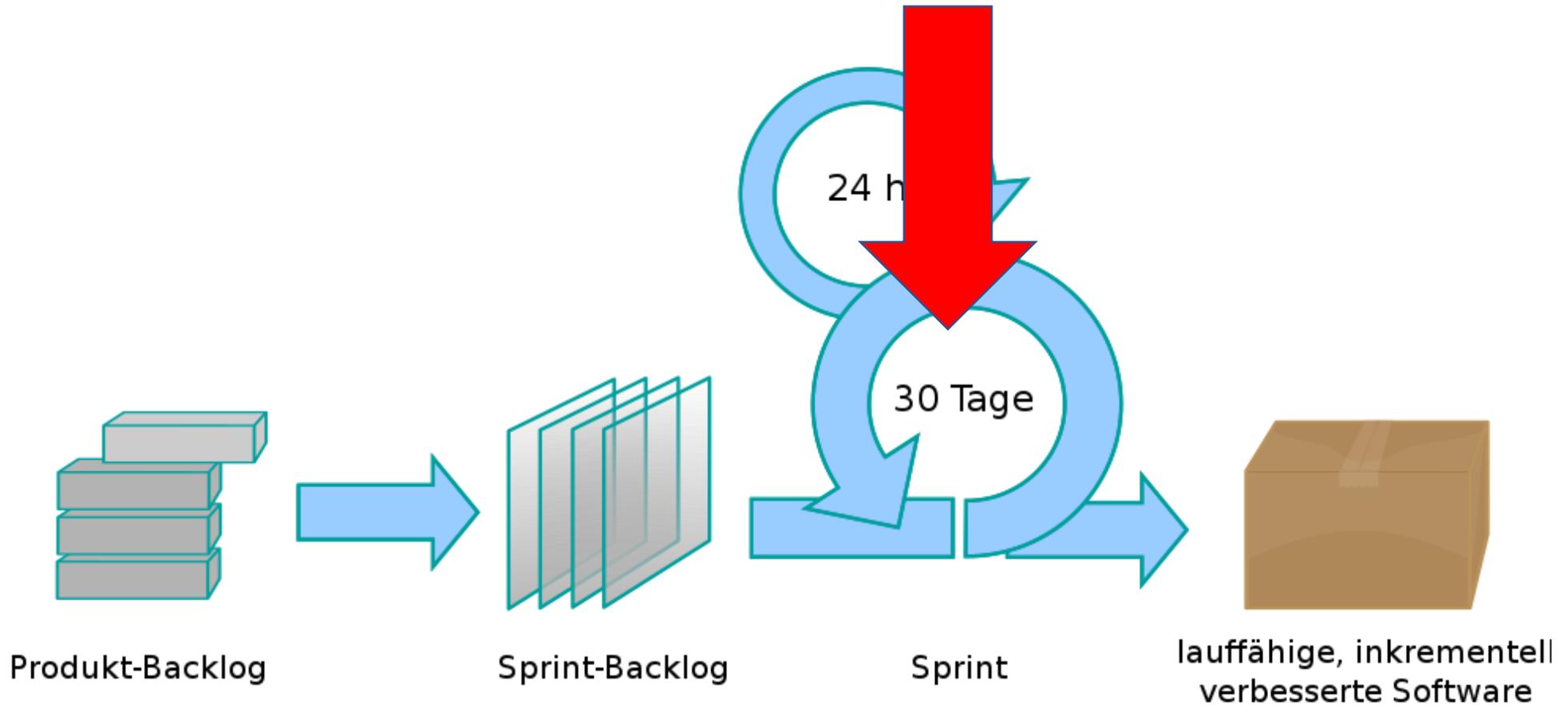


Quelle: Wikipedia

# Lauffähige Software

- Eine neue Version des Projekts
- Lauffähig -> Bereitstellung für den Benutzer
- Besitzt umgesetzte Anforderungen des Sprints

# Refactoring



Quelle: Wikipedia

# Refactoring

- Verbesserung der Softwarequalität
- Rekonstruieren bereits fertiger Implementierungen
- Kein fester Zyklus

# Zusammenfassung

- Der Scrum-Prozess der agilen Softwareentwicklung funktioniert auch bei modellgetriebener Softwareentwicklung
- Unter Verwendung vorhandener XMI-Tools kann auch unter SAP ABAP OO agil und modellgetrieben entwickelt werden
- Der Einsatz beider Technologien zusammen führt am Ende zu langfristig kostengünstigerer und technologisch besserer Software
- Der gemeinsame Ansatz sorgt – bei konsequenter Umsetzung – für einfache und saubere Dokumentationen und eine hohe Kundenzufriedenheit