

Software Management

10. Wiederverwendung

Prof. Dr. K.-P. Fähnrich

30.06.2009

Übersicht der Vorlesung

1. Grundlagen
2. Planung
3. Organisation: Gestaltung
4. Organisation: Prozess-Modelle
5. Personal
6. Leitung
7. Innovationsmanagement
8. Kontrolle: Metriken, Konfigurations- und Änderungsmanagement
9. CASE
10. Wiederverwendung
11. Sanierung

Gliederung

- 1. Zur Problematik**
- 2. Wiederverwendbarkeit und Wiederverwendung**
- 3. Technik**
- 4. Organisation**
- 5. Management**
- 6. Kosten/Nutzen**
- 7. Einführung der Wiederverwendung**

Begleitliteratur: Helmut Balzert, Lehrbuch der Software-Technik
Quelle der Grafiken und Tabellen: Helmut Balzert, Lehrbuch der Software-Technik,
wenn nicht anders angegeben

1. Zur Problematik

„Die Wiederverwendbarkeit von Produktkonzepten, von Produkten aber auch von Verfahrensweisen ist das zentrale technologische Konzept in den hochentwickelten westlichen Industrien. Mit der Wiederverwendung wird sowohl eine Kostensenkung als auch eine Qualitätsverbesserung der Produkte angestrebt.“
[Weber 92]

- Nur durch Wiederverwendung lassen sich:
 - die Produktivität signifikant erhöhen,
 - die Qualität der Produkte verbessern,
 - die Entwicklungszeit kürzen,
 - die Kosten reduzieren.

2. Wiederverwendbarkeit und Wiederverwendung

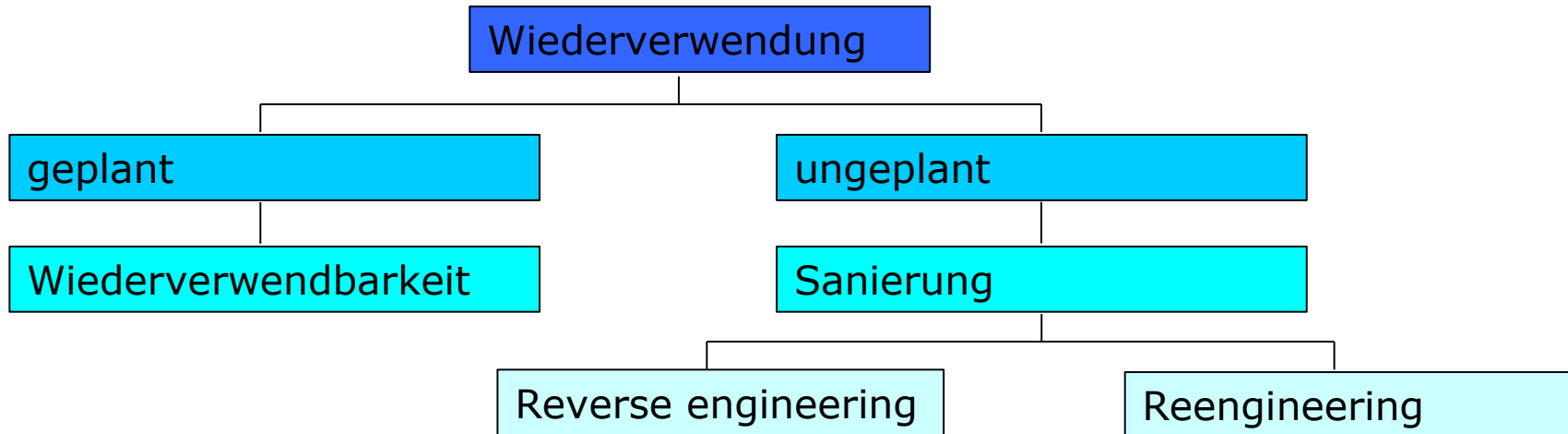
Wiederverwendbarkeit (reusability): Erstellen und Bereitstellen wiederverwendbarer Software.

Wiederverwendung (reuse): Einsatz wiederverwendbarer Software.

- Eine gute Wiederverwendbarkeit und Wiederverwendung setzt eine optimale Zusammenarbeit von
 - Technik,
 - Organisation und
 - Management.voraus.

2. Wiederverwendbarkeit und Wiederverwendung

Arten der Wiederverwendung

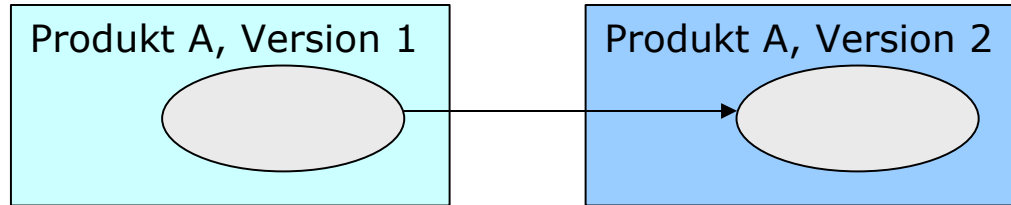


- Üblich ist die ungeplante Wiederverwendung.
- Es wird versucht, Software-Komponente alter Systeme mit Methoden des Reverse Engineering, des Reengineering und der Sanierung zu identifizieren und aufzubereiten.

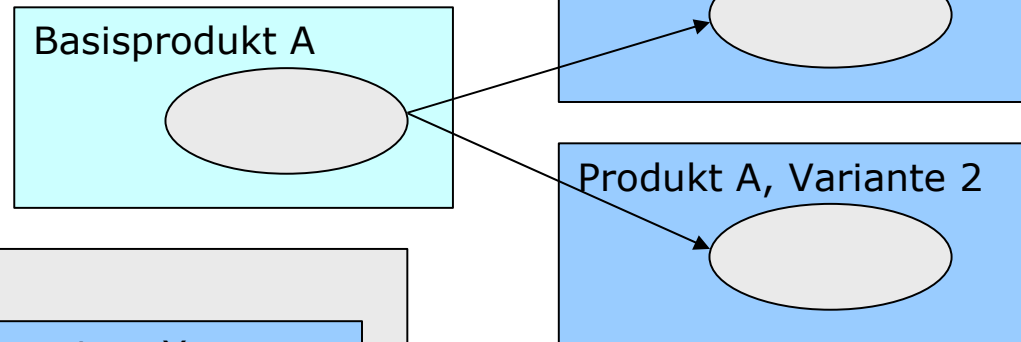
2. Wiederverwendbarkeit und Wiederverwendung

- Bei der geplanten Wiederverwendung geht es darum, Software-Komponenten von vornherein wiederverwendbar so zu gestalten.
- Gute wiederverwendbare Software-Komponenten:
 - besitzen einen hohen Allgemeingrad,
 - sind qualitativ hochwertig und
 - sind gut dokumentiert.
- 4 Typen der Wiederverwendung:
 - Wiederverwendung bei Versionsentwicklung,
 - Wiederverwendung bei Variantenentwicklung,
 - intra-produktorientierte Wiederverwendung,
 - inter-produktorientierte Wiederverwendung.

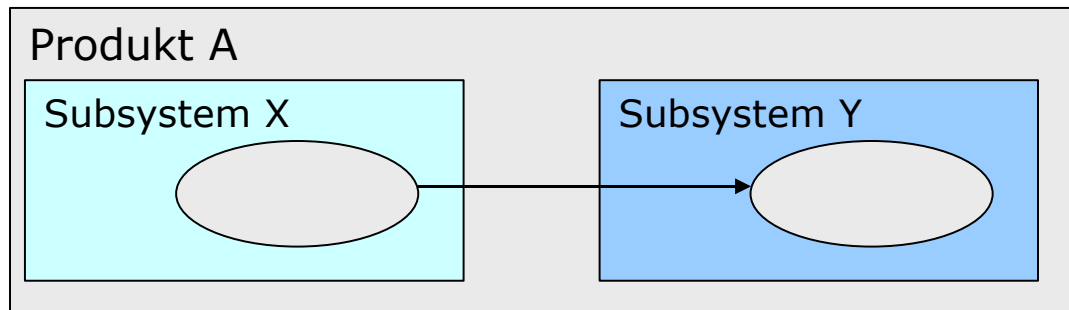
2. Wiederverwendbarkeit und Wiederverwendung



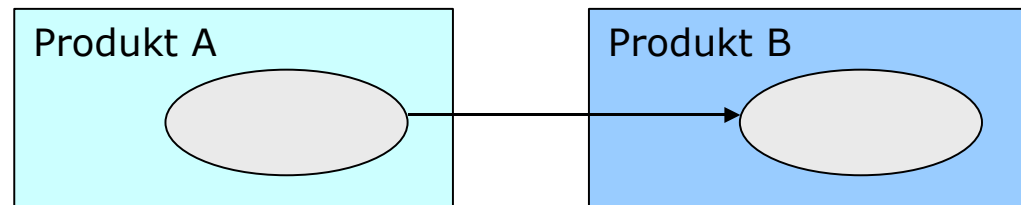
Wiederverwendung bei Versionsentwicklung



Wiederverwendung bei Variantenentwicklung



Intraproduktorientierte Wiederverwendung



Interproduktorientierte Wiederverwendung

2. Wiederverwendbarkeit und Wiederverwendung

- Ebenen der Wiederverwendbarkeit:
 - Produkt-Definition,
 - Produkt-Entwurf,
 - Produkt-Implementierung.
- Differenzierung nach dem Anwendungsbereich:
 - Vertikale Wiederverwendung und
 - Horizontale Wiederverwendung.
- Arten der Wiederverwendung:
 - White-box-Wiederverwendung und
 - Black-box-Wiederverwendung.

Gliederung

- 1. Zur Problematik**
- 2. Wiederverwendbarkeit und Wiederverwendung**
- 3. Technik**
- 4. Organisation**
- 5. Management**
- 6. Kosten/Nutzen**
- 7. Einführung der Wiederverwendung**

Abstraktionsniveaus

- Software-Komponenten besitzen einen unterschiedlichen Grad der Wiederverwendbarkeit in Abhängigkeit ihres Abstraktionsniveaus.
- Abstraktionsniveaus (können alle generisch sein, außer abstrakte Datenobjekte):
 - funktionale Abstraktion,
 - Datenabstraktion,
 - Klassen mit Vererbung und Polymorphismus.

Funktionalen Abstraktion: Charakteristika

Vorteile

- Gut geeignet für einige ausgewählte Anwendungsbereiche

- Abstraktionsniveau zu gering
- Funktionaler Blickwinkel nicht allgemein genug
- Trennung von Wert bzw. Zustand und seiner Bearbeitung
- Unflexibler Parametermechanismus
- Generische, funktionale Abstraktion wird nur von wenigen Programmiersprachen unterstützt.

Nachteile

Datenabstraktion

- In der Datenabstraktion werden Attribute und Operationen zu einer Einheit zusammengefasst.

Vorteile

- Gut geeignet für viele Anwendungsbereiche
- Hoher Allgemeingrad durch Typparametrisierung
- Leicht verständlich

- Nur anwendbar bei streng typisierten Sprachen
- Nicht so allgemein wie die Vererbung

Nachteile

Objektorientierte Komponenten

- Aus Wiederverwendungssicht besitzen Klassen folgende Charakteristika:

Vorteile

- Vererbung erlaubt Spezialisierung durch black-box-Wiederverwendung
- Der Anteil neuen Codes ist minimal, wenn zusätzliche Eigenschaften hinzugefügt werden.

- Bei OO-Systemen wird die systeminhärente Komplexität zum großen Teil auf die Dynamik des Systems, also auf die Kommunikation zwischen den Objekten verlagert
→ Das System ist anhand des Quellcodes schwer zu verstehen.
- Führt zu unübersichtlichen Vererbungshierarchien, wenn

Nachteile

Gliederung

- 1. Zur Problematik**
- 2. Wiederverwendbarkeit und Wiederverwendung**
- 3. Technik**
- 4. Organisation**
- 5. Management**
- 6. Kosten/Nutzen**
- 7. Einführung der Wiederverwendung**

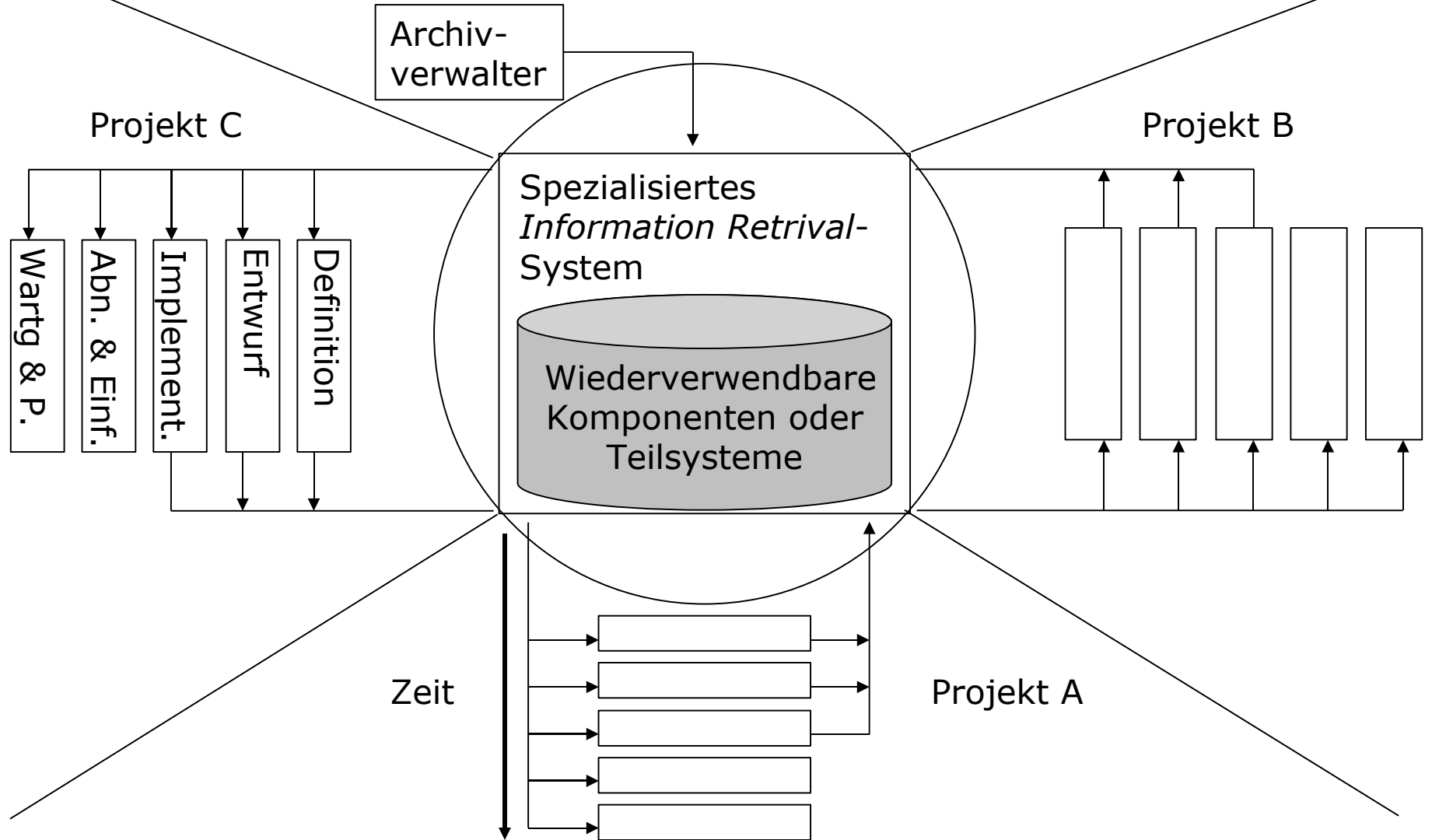
4. Organisation

Organisation in der Wiederverwendung

- Wiederverwendung muss organisiert werden.
- Die Organisation muss folgende Bereiche umfassen:
 - Aufbau, Einrichtung und Betrieb eines Wiederverwendbarkeitsarchivs;
 - Evolutionäre Verbesserung der Wiederverwendung;
 - Einbettung der Wiederverwendung in das Prozessmodell.

4. Organisation

Konzept eines Wiederverwendbarkeitsarchivs



4. Organisation

Wiederverwendbarkeitsarchiv

- Jeder Mitarbeiter soll von seinem Arbeitsplatz über Netz aufs Archiv zugreifen können.
- Ein Wiederverwendbarkeitsarchiv kann:
 - Ein eigenständiges Archiv sein oder
 - Integriert in eine wiederverwendungsorientierte CASE-Umgebung.
- Zum Information Retrieval in einem Wiederverwendbarkeitsarchiv gehört ein geeignetes Dokumentenklassifikationsverfahren (siehe [Zendler 95]).
- Eine Archivverwaltung muss folgende Dienste zur Verfügung stellen:
 - Aufbauen der Klassifikationssysteme,
 - Klassifizieren und Retrieval der Komponenten und Teilsysteme,
 - Export von Komponenten
 - Berichte über Klassifizierungssysteme und über Komponenten

4. Organisation

Wiederverwendungsorientierte CASE-Umgebungen

- Solche CASE-Umgebungen mit integriertem Wiederverwendbarkeitsarchiv müssen zusätzliche Anforderungen erfüllen ([Lindner 96]):
 - Die Verwendung einer Komponente als black-box impliziert die Verwaltung einer Beziehung der Art „nutzt“, um Änderungen an der Originalkomponente der nutzenden mitzuteilen.
 - Die Verwendung einer Komponente als white-box impliziert, dass jede der nutzenden Anwendungen ihre eigene Version aus der Komponente ableiten und weiterentwickeln kann.
 - Unterstützung der Montage von Anwendungen aus Komponenten.
 - Online-Verwaltung aller Ergebnisse aller Werkzeuge während der gesamten Entwicklungszeit.
 - Konfigurations- und Änderungsverwaltung der der wiederverwendbaren Komponenten.

4. Organisation

Evolutionäre Verbesserung der Wiederverwendung [Kauba 97]

Stufe	Wiederverwendung in %	Anforderungen
5 Bereichsbezogene Wiederverwendung	80-100%	Bereichsanalysen und architekturen
4 Konsequente Wiederverwendung	50-70%	Bibliotheken, Prozesse, Metriken, Training
3 Planbare Wiederverwendung	30-40%	Zustimmung und Unterstützung vom Management, Motivationsprogramme, Bibliotheken
2 Ausschachten von Altanwendungen	10-50%	Glück und Wartungsprobleme
1 Keine oder ad hoc Anwendungen	< 20%	Keine organisatorischen Änderungen

4. Organisation

Wiederverwendungsorientiertes Prozessmodell

- Ist eine notwendige Voraussetzung für die Etablierung von Wiederverwendung.
- In allen Entwicklungsphasen wird sowohl nach geeigneten Komponenten gesucht als auch neue wiederverwendbare Komponenten klassifiziert und im Archiv abgelegt.
- Mögliche Quellen von Komponenten:
 - Eigenes Archiv.
 - Externer Komponentenmarkt.
- Beim Kauf von Komponenten müssen folgende Tätigkeiten durchgeführt werden:
 - Auflistung der heutigen und zukünftigen Anforderungen,
 - Prüfen, ob die Komponenten ausreichend parametrisiert sind,
 - Ermitteln ob mit Werkzeugen die Komponenten evaluiert, angepasst und gewartet werden können,
 - Restliche Aspekte der Nutzung der Komponenten klären.

Managementfragen

- Kernfrage des heutigen Managements bei der Entwicklung eines neuen Software-Produkts:
Wo hat bereits jemand ein ähnliches Problem gelöst, und wie bekomme ich die Problemlösung?
- Die Sicherstellung der Wiederverwendung von Komponenten wird in Zukunft eine der Haupttätigkeiten des Managements.
- Die Aufgabe des Software-Managers besteht vor allem darin eine Wiederverwendbarkeits-Kultur zu etablieren.
- Folgende Fragen müssen vom Management beantwortet werden:
 - Wer zahlt für die Entwicklung einer wiederverwendbaren Komponente?
 - Wer übernimmt die Wartungsverpflichtung?
 - Was gewinnt derjenige, die eine wiederverwendbare Komponente zu Verfügung stellt.

Gliederung

- 1. Zur Problematik**
- 2. Wiederverwendbarkeit und Wiederverwendung**
- 3. Technik**
- 4. Organisation**
- 5. Management**
- 6. Kosten/Nutzen**
- 7. Einführung der Wiederverwendung**

6. Kosten/Nutzen der Wiederverwendung

Faustregeln

- *Formel 3* (nach [Lanergan, Grasso 84], [Biggerstaff 94]):
 - Software muss dreimal entwickelt werden, bevor sie wirklich wiederverwendbar entwickelt werden kann.
 - Bevor die Früchte der Wiederverwendung geerntet werden können, muss Software dreimal wiederverwendet werden.
- Nach [Levine 93] sind die Erstellungskosten von wiederverwendbaren Komponenten 60% höher. Diese sind folgendermaßen aufgeteilt:
 - 25% für zusätzliche Verallgemeinerung,
 - 15% für zusätzliche Dokumentation,
 - 10% für zusätzliches Testen,
 - 5% Mehraufwand für Archivablage und Wartung.

6. Kosten/Nutzen der Wiederverwendung

Wiederverwendung und Gesamtproduktivität: Beispiel

	0%	25%	50%
Gesamtzeit in MM	81,5	45	32
Anzahl der Mitarbeiter	8	6	5
Kosten je Zeile	40,75	22,5	16
Zeilen pro MM	165	263	370
Ersparnis	0%	45%	61%

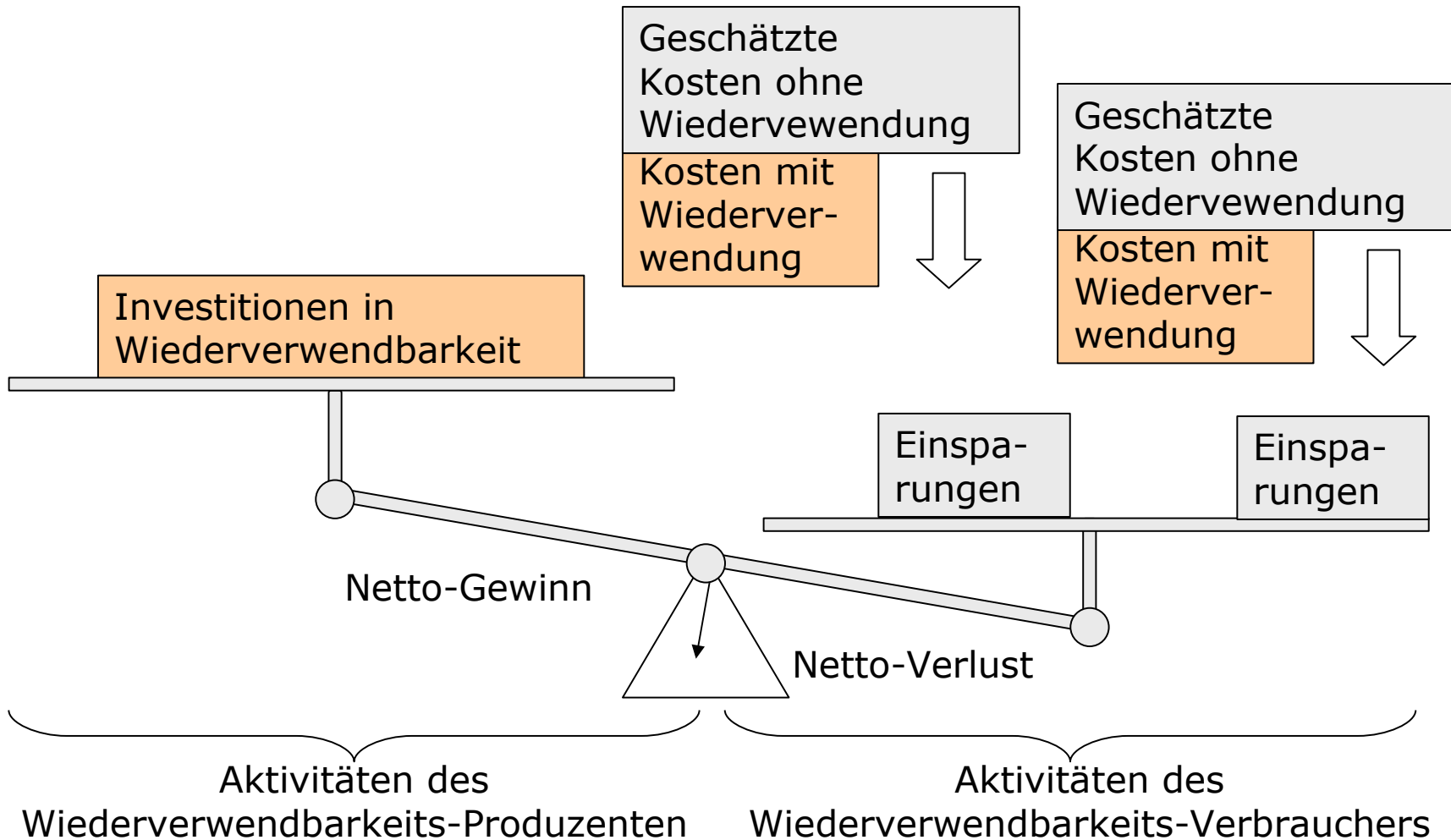
Nach [Jones 86]

	0%	10%	30%	50%	80%
Gesamtzeit in MM	200	176	131	89	33
Ersparnis	0%	12%	34%	56%	84%

Nach [Love 88]

6. Kosten/Nutzen der Wiederverwendung

Kosten/Nutzen Relation der Wiederverwendung



6. Kosten/Nutzen der Wiederverwendung

Return on Investment

- Eine Investition in die Wiederverwendung ist kosteneffektiv wenn $K < N$
wobei K die Gesamtkosten für die Investition sind und N die Einsparungen darstellen.
- Der Nutzen ist folgendermaßen definiert:

$$N = \sum_{i=1}^k n_i = \sum_{i=1}^k (o_i - m_i)$$

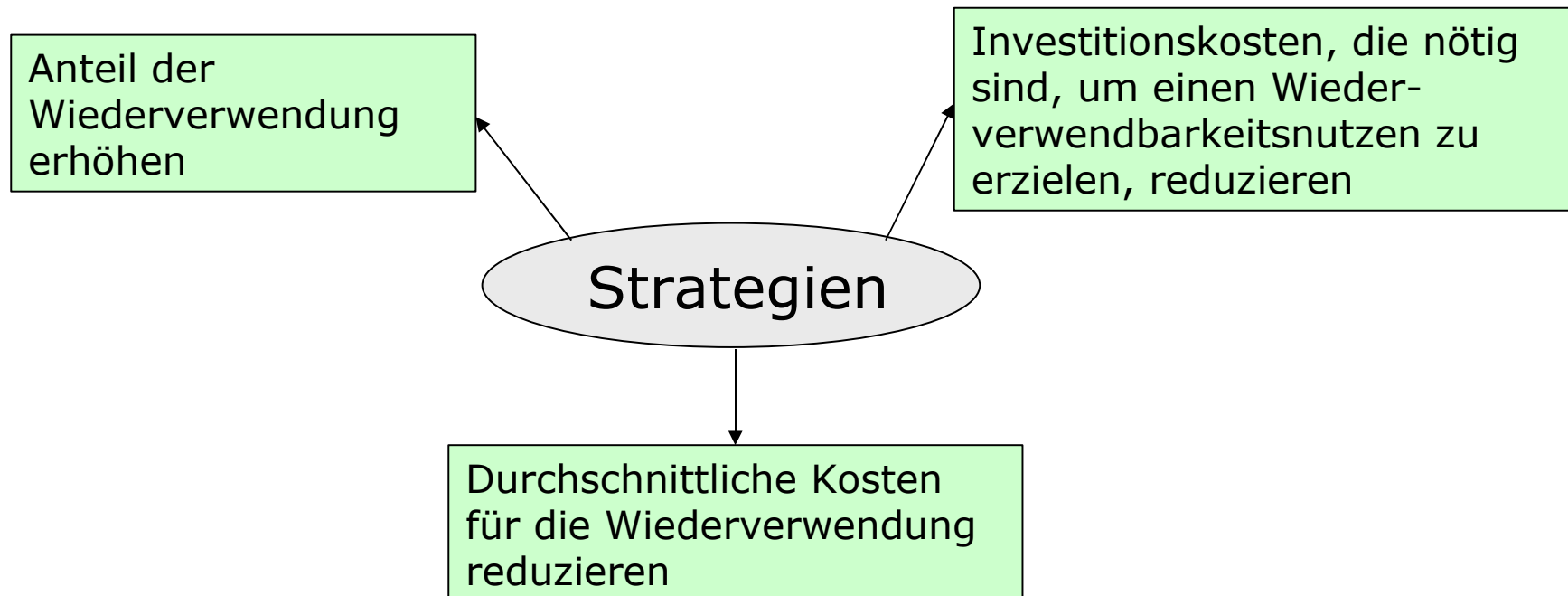
wobei n_i die Einsparungen für die Aktivität i sind, o_i die geschätzten Kosten für die Aktivität i ohne Wiederverwendung und m_i die Kosten für i mit Wiederverwendung. K ist die Anzahl der Aktivitäten die für eine entsprechende Investition tangiert sind.

- Die Kosten/Nutzen Relation (auch Return-on-investment , RoI) genannt ergibt sich zu:
 $R = N/K$
- $R < 1 \rightarrow$ Verlust

6. Kosten/Nutzen der Wiederverwendung

Kosteneffektive Wiederverwendung

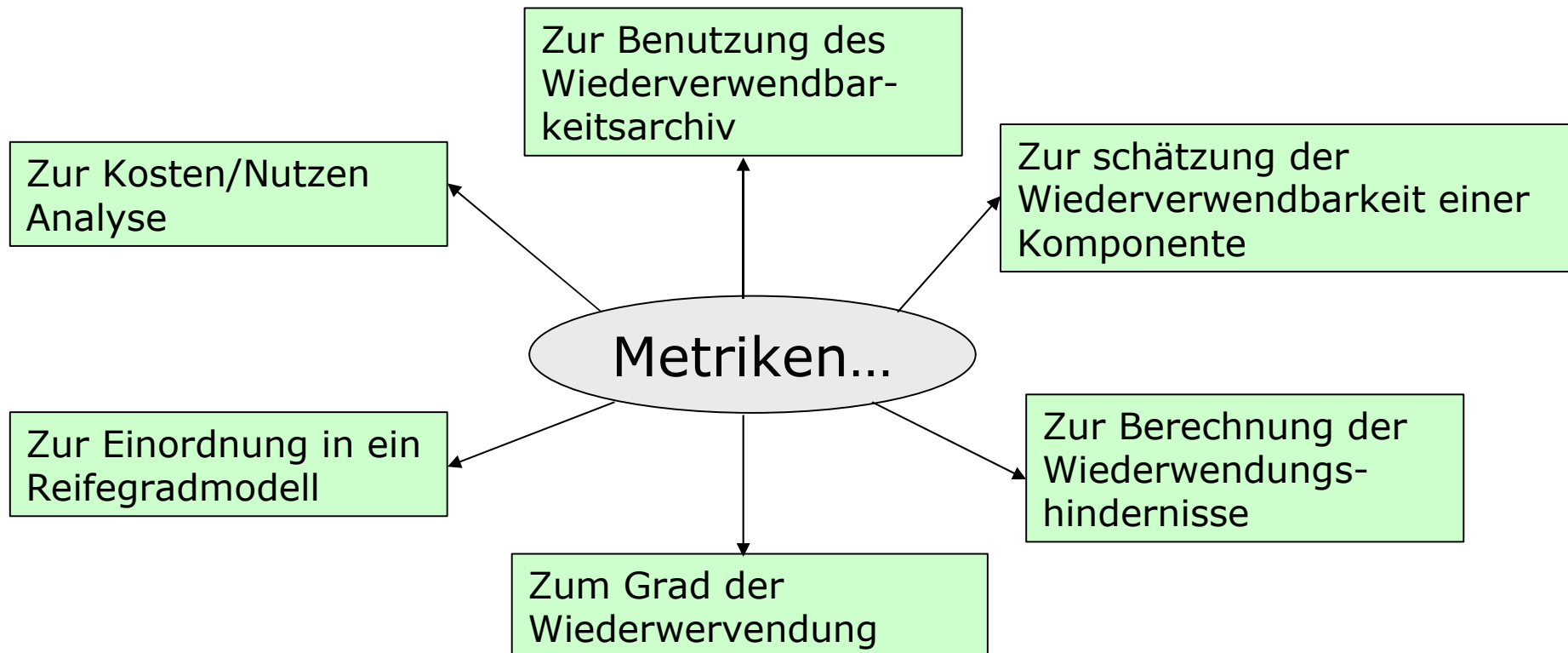
- Um die Wiederverwendung kosteneffektiv zu machen, muss R möglichst groß gemacht werden.
- 3 mögliche Strategien:



6. Kosten/Nutzen der Wiederverwendung

Kosteneffektive Wiederverwendung (2)

- 6 mögliche Metrikkategorien zum Messen verschiedener Aspekte der Wiederverwendung nach [Frakes, Terry 96].



7. Einführung der Wiederverwendung

Einführung & Hindernisse

- Orientiert man sich an ein Reifegradmodell, dann besteht die erste Aufgabe darin, die Ist-stufe zu ermitteln.
- Anschließend ist dann eine Planung aufzustellen, wie man schrittweise die jeweils nächste Stufe erreicht.
- Erfahrungen haben gezeigt, dass Anforderungen an Komponenten – zumindest am Anfang- nicht zu hoch sein dürfen ([Kauba 97]).
- Wesentlich für die Akzeptanz von Wiederverwendung ist ein überzeugtes Management, das Wiederverwendung als Mittel zur Erleichterung der allgemeinen Unternehmensziele betrachtet.
- Wiederverwendung impliziert einen Wandel der Entwicklungstätigkeit: vom schreibenden Entwickler zum lesenden, evaluierenden und kreativ komponierenden Software-Architekten.

7. Einführung der Wiederverwendung

Einführung & Hindernisse (2)

Potentielle Hindernisse bei der Einführung (nach [Kauba 97])

<p style="text-align: center;">Ökonomisch</p> <ul style="list-style-type: none">• fehlendes Commitment• unklare Geschäftsstrategie• Investitionshöhe• Aufwandsgeschäft• fehlende Nutzungs- und Verwertungsrechte.	<p style="text-align: center;">Organisatorisch</p> <ul style="list-style-type: none">• Im Prozess nicht vorgesehen• Verantwortung nicht zugewiesen• Fehlender Katalysator• Fehlende Infrastruktur
<p style="text-align: center;">Soziologisch</p> <ul style="list-style-type: none">• Not-invented-here-Syndrom• Widerstand gegen Veränderungen• Existenzängste• Selbstverständnis des Entwicklers/ geändertes Rollenbild	<p style="text-align: center;">Technisch</p> <ul style="list-style-type: none">• Fehlende Erfahrung mit praktischen Anwendungen• Mangelndes Know-How• Schwächen im SW-Engineering-Prozess• Fehlende Tools

- [Biggerstaff 94]
Biggerstaff T.J., Is Technology a Second Order Term in reuse's Success Equation, in Proceedings of Third International Conference on Software Reuse
- [Frakes, Terry 96]
Frakes W., Terry C., Software Reuse: Metrics and Models, ACM Computing Survey
- [Jones 86]
Jones C., the Impact of Reusable Models and Functions, Mc Grow-Hill
- [Kauba 97]
Kauba, E., Wiederverwendung als Gesamtkonzept, Objektspektrum
- [Lanergan, Grass 84]
Lanergan R.G., Grasso C.A., Software Engineering with Reusable Design and Code, in IEEE Transactions on Software Engineering.
- [Lindner 96]
Lindner U., Massive Wiederverwendung, Objektspektrum
- [Levine 93]
Levine T., Reusable Software Components, ACM Ada Letters
- [Love 88]
Love T., The Economics of Reuse, IEEE COMPCON
- [Weber 92]
Weber H., Die Software-Krise und ihre Macher, Springer-Verlag
- [Zendler 95]
Zendler A., Konzepte, Erfahrungen und Werkzeuge zur Software-Wiederverwendung, Tectum-verlag.