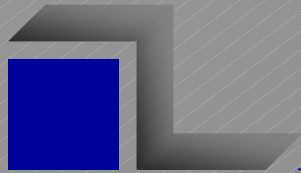


# Anforderungsanalyse, *Requirements Engineering*

---

- Begriffe
  - Anforderungen, Lastenheft, Pflichtenheft, Spezifikation
- Notation, Zielgruppen
  - Natürliche Sprache, Formulare
  - Pflichtenheft, Anforderungen an ein Pflichtenheft
  - Beispiel
- Typen von Anforderungen  
Funktionale, nicht-funktionale und Problembereichs-Anforderungen
- Prozesse
- Management von Anforderungen
- Die sieben Sünden...
- Auszug: *Der Termin*

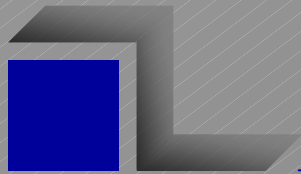


# Frage

## Anforderungen

- ▶ Begriffe
- ▶ Notation
- ▶ Beispiel
- ▶ Anforderungs-Typen
- ▶ Prozesse
- ▶ Management
- ▶ Seven Sin

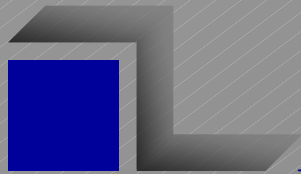
Welche drei (zwei)  
"Dinge" müssen durch  
Anforderungen an ein  
System festgelegt  
werden?



## Anforderungen

- ▶ Begriffe
- ▶ Notation
- ▶ Beispiel
- ▶ Anforderungs-Typen
- ▶ Prozesse
- ▶ Management
- ▶ Seven Sin

- **Anforderungen** an ein System beschreiben
  - die Dienste, die ein Kunde von einem System erwartet, und
  - die Gegebenheiten, unter denen es entwickelt wird und laufen soll
  
- **Anforderungsanalyse** (*Requirements Engineering*) heißt der Prozess des
  - Herausfindens (Elicitation),
  - Analysierens (Analysis),
  - Dokumentierens (Specification) und
  - Überprüfens (Validation)dieser Anforderungen



# Detailgrad

Anforderungen

▶ Begriffe

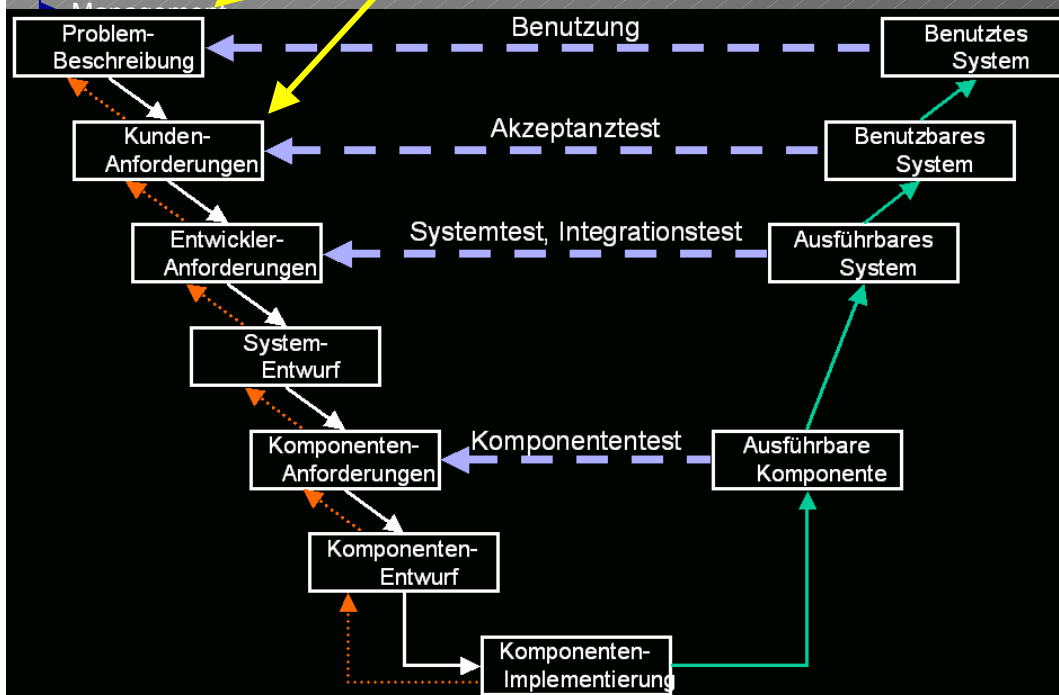
▶ Notation

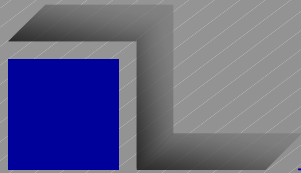
▶ Beispiel

▶ Anforderungs-  
Typen

▶ Prozesse

- Anforderungen können sehr abstrakt oder sehr detailliert sein (bis hin zu funktionalen Spezifikationen)
- Anforderungen erfüllen zwei Aufgaben
  - Basis für eine **Ausschreibung** für eine Software:  
Die Erledigung der Aufgaben muss **interpretierbar** sein
  - Basis für einen **Vertrag** über eine Software-Entwicklung:  
Die Aufgaben der Software müssen **detailliert festgelegt** sein





# Begründung für den unterschiedlichen Detailgrad

[Davis '93]

Anforderungen

▶ Begriffe

▶ Notation

▶ Beispiel

▶ Anforderungs-  
Typen

▶ Prozesse

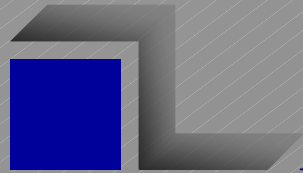
▶ Management

▶ Seven Sin

- Wenn ein Unternehmen einen Auftrag für ein großes Softwareentwicklungsprojekt vergeben möchte, muss es dessen Erfordernisse in einem **Lastenheft** in einer so abstrakten Form definieren, dass der Lösung nicht vorgegriffen wird.

Die Anforderungen müssen so beschrieben werden, dass sich mehrere Interessenten um den Zuschlag bewerben können, die sich zum Beispiel durch verschiedene Lösungsansätze unterscheiden.

- Ist erst einmal ein Vertrag zu Stande gekommen, muss der Vertragspartner (Auftragnehmer) eine genauere Systemdefinition für den Kunden aufstellen (**Pflichtenheft**), so dass der Kunde verstehen und beurteilen kann, was die Software tun wird.
- Diese beiden Dokumente können das Anforderungsdokument (**Requirements Document**) des Systems genannt werden.



# Partner-Diskussion: Verwendung von Anforderungsdokumenten

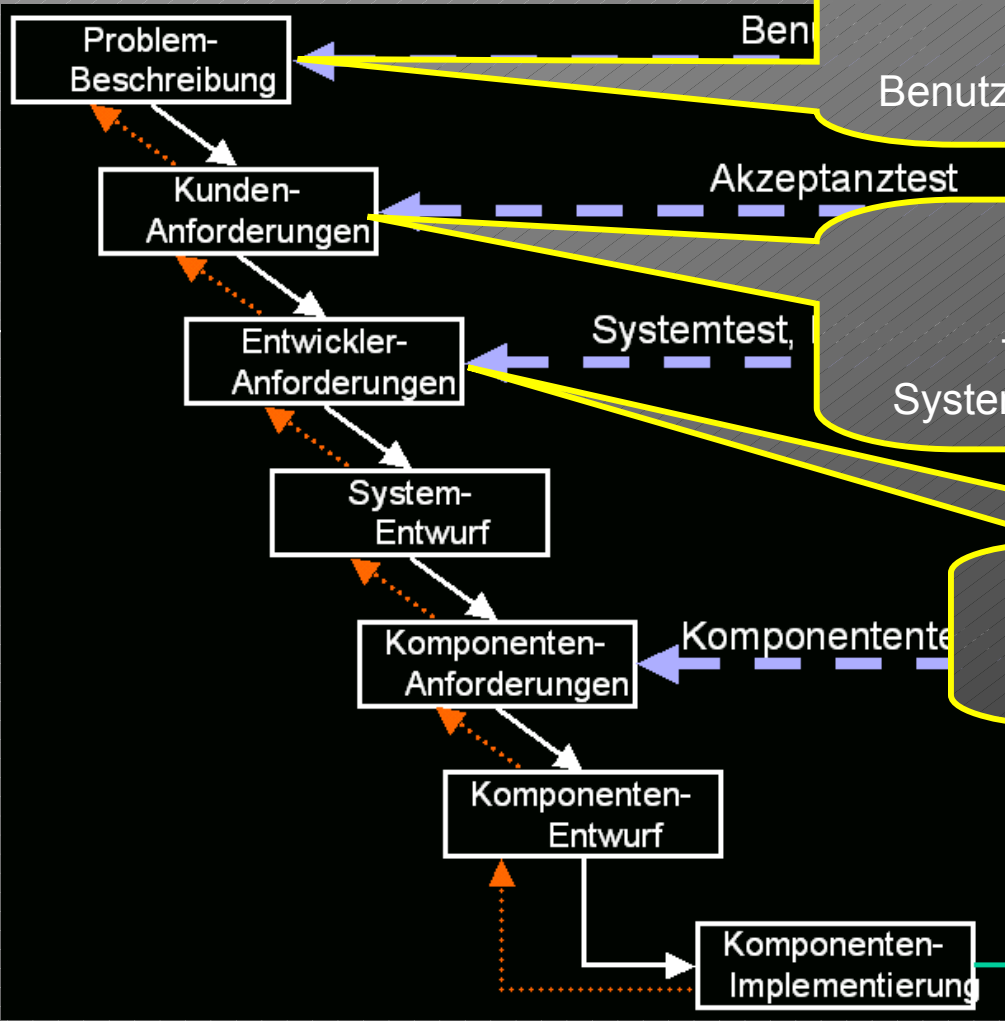
## Anforderungen

- ▶ Begriffe
- ▶ Notation
- ▶ Beispiel
- ▶ Anforderungs-  
Typen
- ▶ Prozesse
- ▶ Management
- ▶ Seven Sin

- Diskutieren Sie mit einem Partner
  - Welche Vorteile hat die Tatsache, dass Anforderungen (zwei) unterschiedliche Aufgaben erfüllen (müssen)?
  - Welche Nachteile ergeben sich aus dieser Tatsache?
- Dauer: 3 Minuten

# Begriffsübersicht

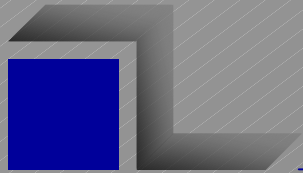
- Anforderungen
- ▶ Begriffe
- ▶ Notation
- ▶ Beispiel
- ▶ Anforderungs-Typen
- ▶ Prozesse
- ▶ Management
- ▶ Seven Sin



Ausschreibung:  
Lastenheft,  
Benutzeranforderungen [Sommerville]

Vertragsgrundlage:  
Pflichtenheft,  
funktionale Spezifikation,  
Systemanforderungen [Sommerville]

Reproduktion des  
Pflichtenhefts für den  
Entwickler-internen Gebrauch

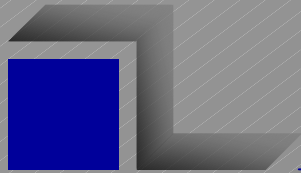


## Anforderungen

- ▶ Begriffe
- ▶ Notation
- ▶ Beispiel
- ▶ Anforderungs-Typen
- ▶ Prozesse
- ▶ Management
- ▶ Seven Sin

# Fragen?





## Anforderungen

- ▶ Begriffe
- ▶ Notation
- ▶ Beispiel
- ▶ Anforderungs-Typen
- ▶ Prozesse
- ▶ Management
- ▶ Seven Sin

- Welche Notationsform für Anforderungen können Sie sich vorstellen?
- Welche Vorteile haben diese jeweils?
- Welche Nachteile haben diese jeweils?

# Übersicht über mögliche Notationsformen

4 Aspekte müssen festgelegt werden

	Schnittstelle	Verhalten	Struktur	Ablauf
Anforderungen	X	X	X	X
Begriffe	X	X		
Notation	X	X	X	X
Beispiel	X			
Anforderungs-Typen	X	X		X
Prozesse			X	X
Management			X	
Seven Sin				
Natürliche Sprache	X	X		
Standardisierte Formulare	X	X		
Standardisierte Dokumente	X	X		
Formale Sprachen	X			
Datenflussgraphen	X	X		X
Struktogramme			X	X
Entity-Relationship- Diagramme			X	
Logische Spezifikationen	X	X		
Funktionale Spezifikationen	X	X		
Axiomatische Spezifikationen		X		
Entscheidungstabellen		X		
Zustandstabellen		X		
Petri-Netze		X		
UML	X	X	X	X

# Natürliche Sprache: Beispiel Lastenheft

## Anforderungen

▶ Begriffe

▶ Notation

▶ Beispiel

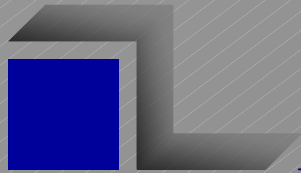
▶ Anforderungs-  
Typen

▶ Prozesse

▶ Management

▶ Seven Sin

- *Meetings* (Treffen) werden typischerweise in der folgenden Art und Weise arrangiert. Ein Meeting-Initiator fragt alle potenziellen Teilnehmer nach den folgenden Informationen, basierend auf ihrer persönlichen Zeitplanung:
  - einer Menge von Zeiten, zu denen sie nicht an dem Meeting teilnehmen können (nachfolgend als *Ausschlusszeiten* bezeichnet);
  - einer Menge von Zeiten, zu denen sie bevorzugt an dem Meeting teilnehmen möchten (nachfolgend als *Vorzugszeiten* bezeichnet).
- Eine Meeting-Zeit wird durch ein Paar (Kalenderdatum, Zeitdauer) beschrieben. Die Ausschluss- und Vorzugszeiten liegen alle in einem vom Meeting-Initiator vorgegebenen Zeitintervall (nachfolgend als *Zeitraum* bezeichnet).
- Der Meeting-Initiator fragt die aktiven Teilnehmer auch nach speziellen Ausstattungswünschen für den Meeting-Raum (z.B. Overhead-Projektor, Rechner, Netzwerkanbindung, Telefon); er/sie kann auch wichtige Teilnehmer nach ihrem bevorzugten Raum für das Meeting fragen.

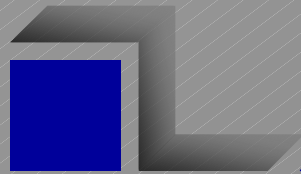


# Natürliche Sprache: Vor- und Nachteile

## Anforderungen

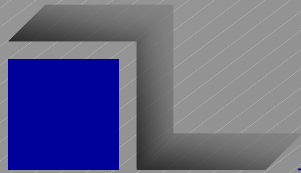
- ▶ Begriffe
- ▶ Notation
- ▶ Beispiel
- ▶ Anforderungs-Typen
- ▶ Prozesse
- ▶ Management
- ▶ Seven Sin

- Vorteile
  - Leicht verständlich
  - Hilfreich für eine oberflächliche, einführende Beschreibung
- Nachteile
  - Mehrdeutig
  - Keine Möglichkeit, Vollständigkeit und Konsistenz (automatisch) zu überprüfen
  - Keine Möglichkeit, formale Verifikation oder Beweise anzuwenden
- Aufgabe
  - Suchen Sie eventuelle
    - Unklarheiten
    - Mehrdeutigkeiten
    - Widersprücheim Beispiel-Text



# Standard-Formular: Beispiel Entwickler-Anforderung

<b>Funktion</b>	Geld vom Konto abheben, withdraw
<b>Beschreibung</b>	Ein Geldbetrag in einer gegebenen Wahrung wird von einem Konto abgebucht, wenn das Konto die entsprechende Deckung aufweist
<b>Eingaben</b>	accountNo : long; die Kontonummer amount : float; der Betrag, der abgebucht werden soll currency : String; die Wahrung, in der abgehoben werden soll (3-Buchstaben-Kurzel)
<b>Herkunft</b>	Kontonummer, Betrag und Wahrung werden vom Benutzer eingegeben
<b>Ausgabe</b>	Ein Boole'scher Wert, der angibt, ob die Abhebung moglich war
<b>Ziel</b>	Die graphische Programmoberflache (GUI), die dem Benutzer meldet, ob das Abheben erfolgreich war
<b>Benotigt</b>	uberpruft beim Converter, ob die Wahrung currency bekannt ist; lasst den Betrag amount vom Converter in den aquivalenten Euro-Betrag euroAmount umrechnen; uberpruft bei der AccountList, ob ein Konto account mit der Nummer accountNo existiert; fragt von der AccountList den Kontostand des Kontos account ab; lasst von der AccountList den Betrag euroAccount vom Konto account abbuchen
<b>Seiteneffekte</b>	Das Konto mit der Kontonummer accountNo
<b>Vorbedingung</b>	---
<b>Nachbedingung</b>	Wenn amount $\geq 0$ und currency bekannt ist und accountNo existiert und account.balance $\geq$ euroAmount dann wurde account.balance um euroAmount vermindert



# Pflichtenheft nach IEEE/ANSI 830-1993

Anforderungen

- ▶ Begriffe
- ▶ Notation
- ▶ Beispiel
- ▶ Anforderungs-Typen
- ▶ Prozesse
- ▶ Management
- ▶ Seven Sin

## 1. Einleitung

1. Ziel des Anforderungsdokuments
2. Anwendungsbereich des Produkts
3. Definitionen, Akronyme und Abkürzungen
4. Referenzen
5. Überblick über den Rest des Dokuments

## 2. Allgemeine Beschreibung

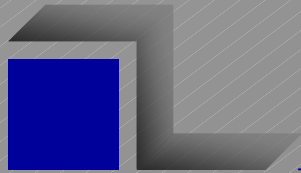
1. Produktperspektive
2. Produktfunktionen
3. Benutzercharakteristika
4. Allgemeine Beschränkungen
5. Voraussetzungen und Abhängigkeiten

## 3. Spezifische Anforderungen

Funktionale, nicht-funktionale und Schnittstellen-Anforderungen

## 4. Anhänge

## 5. Index



# Pflichtenheft nach DIN 66901

Anforderungen

▶ Begriffe

▶ Notation

▶ Beispiel

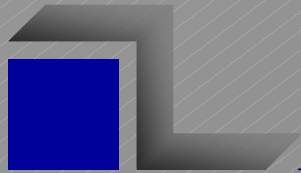
▶ Anforderungs-  
Typen

▶ Prozesse

▶ Management

▶ Seven Sin

1. Zielbestimmung
  1. Muss-Kriterien
  2. Wunsch-Kriterien
  3. Abgrenzungskriterien
2. Produkt-Einsatz
  1. Anwendungsbereiche
  2. Zielgruppen
  3. Betriebsbedingungen
3. Produkt-Bedingungen
  1. Software
  2. Hardware
  3. Orgware
  4. Produkt-Schnittstellen
4. Produkt-Funktion
5. Produkt-Leistungen
6. Benutzer-Schnittstelle
7. Qualitäts-Zielbestimmung
8. Globale Testfälle
9. Ergänzungen



# Anforderungen an ein Pflichtenheft 1/2

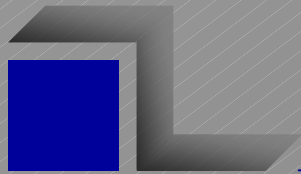
## Anforderungen

- ▶ Begriffe
- ▶ Notation
- ▶ Beispiel
- ▶ Anforderungs-Typen
- ▶ Prozesse
- ▶ Management
- ▶ Seven Sin

## Heninger:

- Es sollte nur das äußere Systemverhalten festlegen
- Es sollte Beschränkungen bezüglich der Implementierung vermeiden
- Es sollte leicht verständlich sein
- Es sollte als Referenz für Systemwarter dienen (können)
- Es sollte Vorüberlegungen zum Lebenszyklus des Systems enthalten
- Es sollte akzeptable Reaktionen auf potenzielle Systemfehler beschreiben





# Anforderungen an ein Pflichtenheft 2/2

## Anforderungen

- ▶ Begriffe
- ▶ Notation
- ▶ Beispiel
- ▶ Anforderungs-Typen
- ▶ Prozesse
- ▶ Management
- ▶ Seven Sin

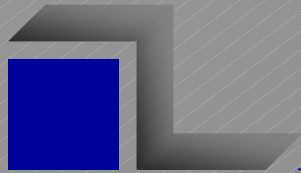
## Sommerville:

Das Pflichtenheft ist kein Design-Dokument. Soweit irgend möglich, sollte es nur festlegen, **WAS** das System tun soll, **NICHT**, **WIE** es das tun soll

## Rombach:

Empirisch belegt ist die Relevanz der vier C's

- Clarity
- Completeness
- Consistency
- Correctness



# Zielgruppen für ein Pflichtenheft

Anforderungen

▶ Begriffe

▶ Notation

▶ Beispiel

▶ Anforderungs-  
Typen

▶ Prozesse

▶ Management

▶ Seven Sin

Systemkunden

- Festlegen von Anforderungen
- Überprüfen, ob die Anforderungen geeignet sind
- Verantwortlich für Änderungen

Manager

- Erstellen des Angebots
- Planung des Entwicklungsprozesses

Systementwickler

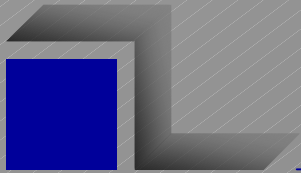
- Verstehen, was für ein System entwickelt werden soll

Systemtester

- Entwickeln von Validierungstests

Systemwarter

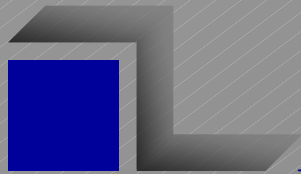
- Verstehen des Systems und der Beziehungen zwischen seinen Bestandteilen



## Anforderungen

- ▶ Begriffe
- ▶ Notation
- ▶ Beispiel
- ▶ Anforderungs-Typen
- ▶ Prozesse
- ▶ Management
- ▶ Seven Sin

# Fragen?



# Beispiel: Lastenheft und Pflichtenheft

## Anforderungen

▶ Begriffe

▶ Notation

▶ Beispiel

▶ Anforderungs-  
Typen

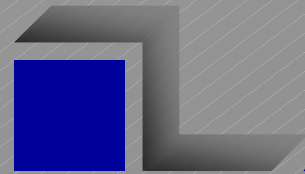
▶ Prozesse

▶ Management

▶ Seven Sin

1. Die Software muss über Mittel zur Darstellung externer, von anderen Werkzeugen erzeugter Dateien verfügen und auf sie zugreifen können.

- 1.1 Der Benutzer sollte über Möglichkeiten zur Definition externer Dateitypen verfügen.
- 1.2 Jeder externe Dateityp kann eine damit verknüpfte Anwendung besitzen, mit der die Datei bearbeitet wird.
- 1.3 Jeder externe Dateityp kann als bestimmtes Symbol auf dem Bildschirm des Anwenders dargestellt werden.
- 1.4 Es sollten Möglichkeiten zur Definition des externen Symbols für externe Dateitypen durch den Benutzer bereitgestellt werden.
- 1.5 Wenn ein Benutzer ein Symbol auswählt, das eine externe Datei repräsentiert, so soll die durch dieses Symbol dargestellte Datei mit der Anwendung geöffnet werden, die mit dem entsprechenden externen Dateityp verknüpft ist.



## Beispiel: Entwickler-Anforderung

<b>Funktion</b>	Externe Datei öffnen, openExternalFile
<b>Beschreibung</b>	Zu einem ausgewählten Symbol wird die Datei identifiziert, die durch dieses Symbol repräsentiert wird und die Anwendung, die mit diesem Dateityp verknüpft ist; dann wird die Datei mit der Anwendung geöffnet.
<b>Eingaben</b>	symbol : DesktopSymbol; das ausgewählte Symbol mit Dateiname filename
<b>Herkunft</b>	Das Symbol wird vom Benutzer ausgewählt
<b>Ausgabe</b>	---
<b>Ziel</b>	---
<b>Benötigt</b>	Eine Typ-Tabelle, in der die zum Dateityp filetype gehörige Anwendung application identifiziert wird
<b>Seiteneffekte</b>	Ein Fenster wird geöffnet, in dem die Anwendung application mit der Datei filename gestartet wird
<b>Vorbedingung</b>	Der Dateiname filename besitzt einen Suffix bestehend aus drei Zeichen (dadurch wird der Dateityp filetype identifiziert)
<b>Nachbedingung</b>	---

<b>Funktion</b>	Anwendung zu einem Dateityp identifizieren, getApplicationForFileType
<b>Beschreibung</b>	Zu einem Dateityp wird die zugehörige Anwendung identifiziert; wurde mit einem Typ keine Anwendung verknüpft, dann wird der Standard-Editor geöffnet
<b>Eingaben</b>	filetype : String; der Dateityp, der durch drei Zeichen repräsentiert wird
<b>Herkunft</b>	Der filetype wird vom Datei-Manager extrahiert
<b>Ausgabe</b>	application : String; Pfadname der Anwendung, die mit diesem Dateityp verknüpft ist oder Pfadname des Standard-Editors
<b>Ziel</b>	Datei-Manager (der die entsprechende Anwendung öffnen möchte)
<b>Benötigt</b>	---
<b>Seiteneffekte</b>	---
<b>Vorbedingung</b>	---
<b>Nachbedingung</b>	---