

# SoftWert

BMBF-Projekt zur Entwicklung eines Methodenbaukastens  
zur Verwertung von Forschungssoftware

## Innofo-Projektvorstellung "SoftWert"

*Zahra Saleh, Dr. Janine Fischer, Lisa Wenzel, Michael Jung*

*Online-Präsentation, 06.03.2024*



# Agenda

TOP	Thema	Referent
1	Begrüßung und Einleitung	Zahra Saleh
2	<b>Der Methodenbaukasten – die Inhalte</b>	<b>Zahra Saleh</b>
3	Fokusthema 1: Rechtliche Fallstricke	Janine Fischer
<b>15 min Pause</b>		
4	Fokusthema 2: Innovative Transferwege	Lisa Wenzel Michael Jung
5	Good Practice Beispiele aus den Zentren der Projektbeteiligten	Lisa Wenzel Janine Fischer
6	Software Dealdatenbank	Zahra Saleh
7	Take Home Messages	Lisa Wenzel
8	Diskussion und Fragen	alle
<b>Ende des Meetings</b>		

# SoftWert

Entwicklung und  
Implementierung eines  
Methodenbaukastens zur  
Verwertung von  
Forschungssoftware

## PROJEKTPARTNER

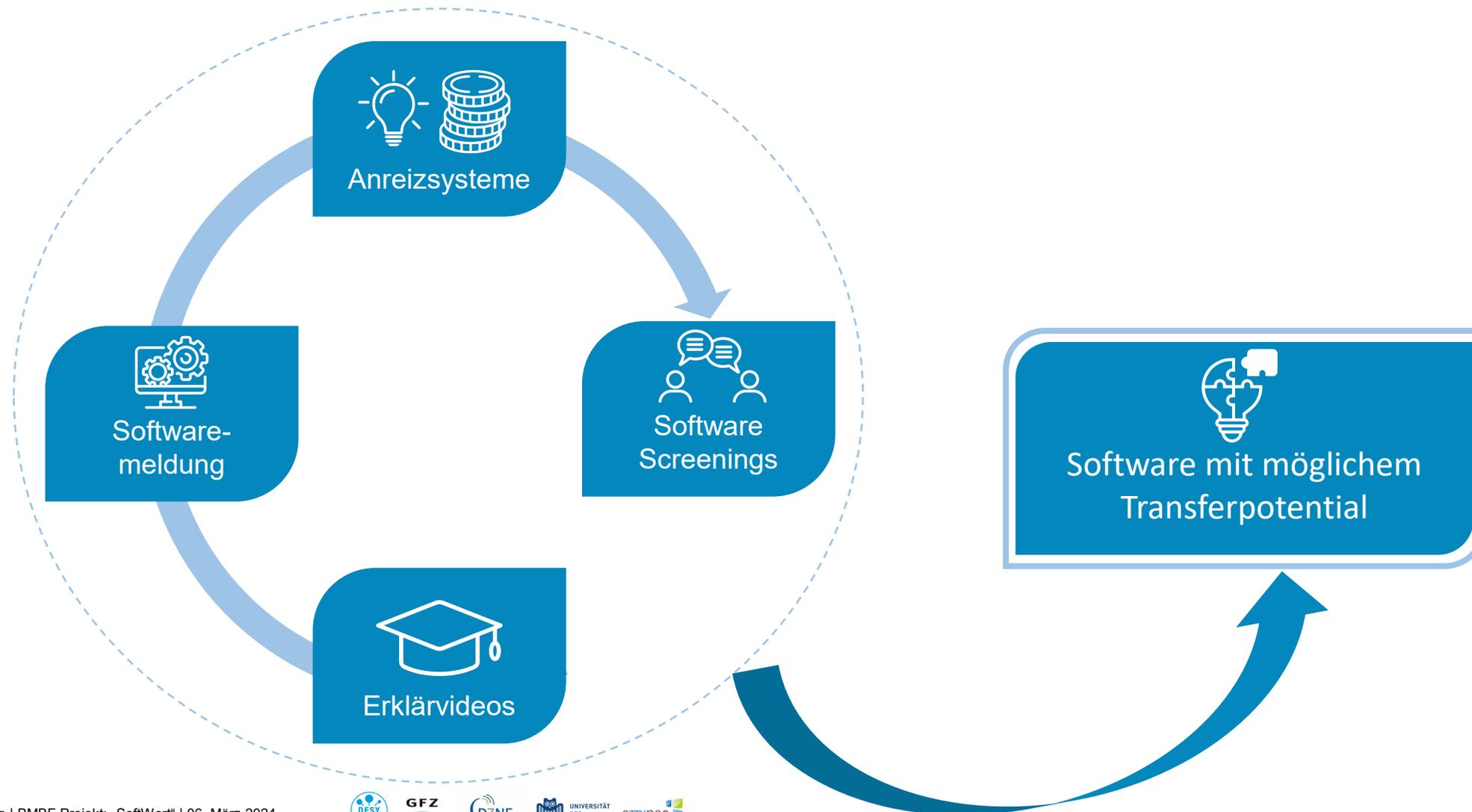
- ❖ **Forschungssoftware = wertvolle Ressource**
- ❖ **Bedarf an Good Practices für Transfer**
- ❖ **Der Methodenbaukasten:  
Lösungsansätze und Tools für Softwareverwertung**



# Der Methodenbaukasten

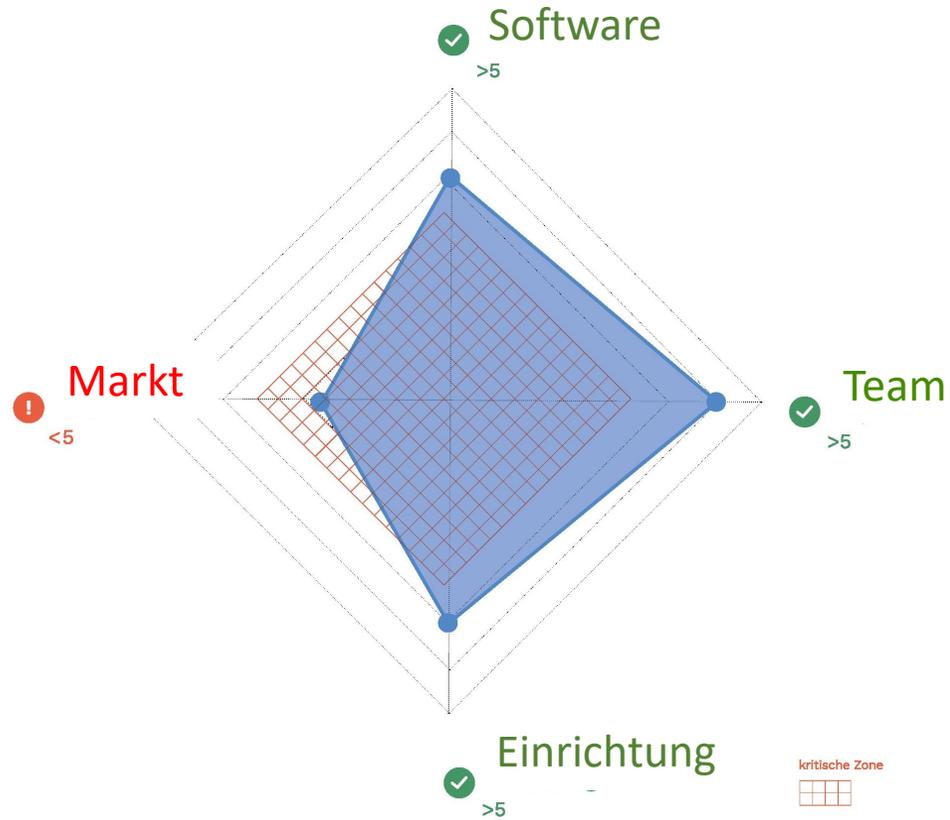


# Sensibilisierung & Erfassung von Forschungssoftware



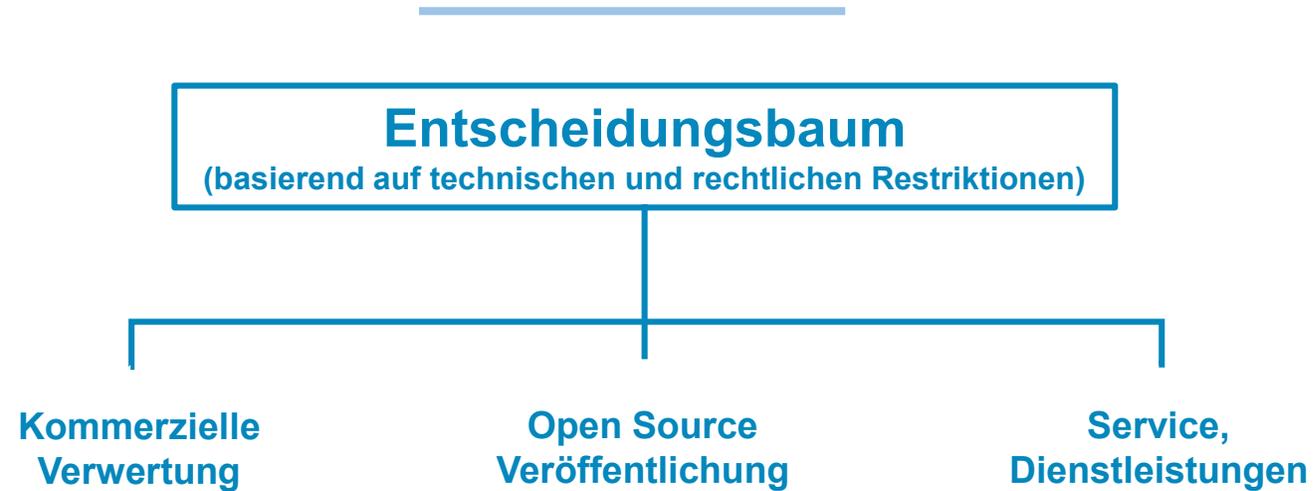
# Bewertung des Verwertungspotentials

## Das Bewertungstool



Handlungsempfehlungen

# Entscheidungshilfe für Transferwege

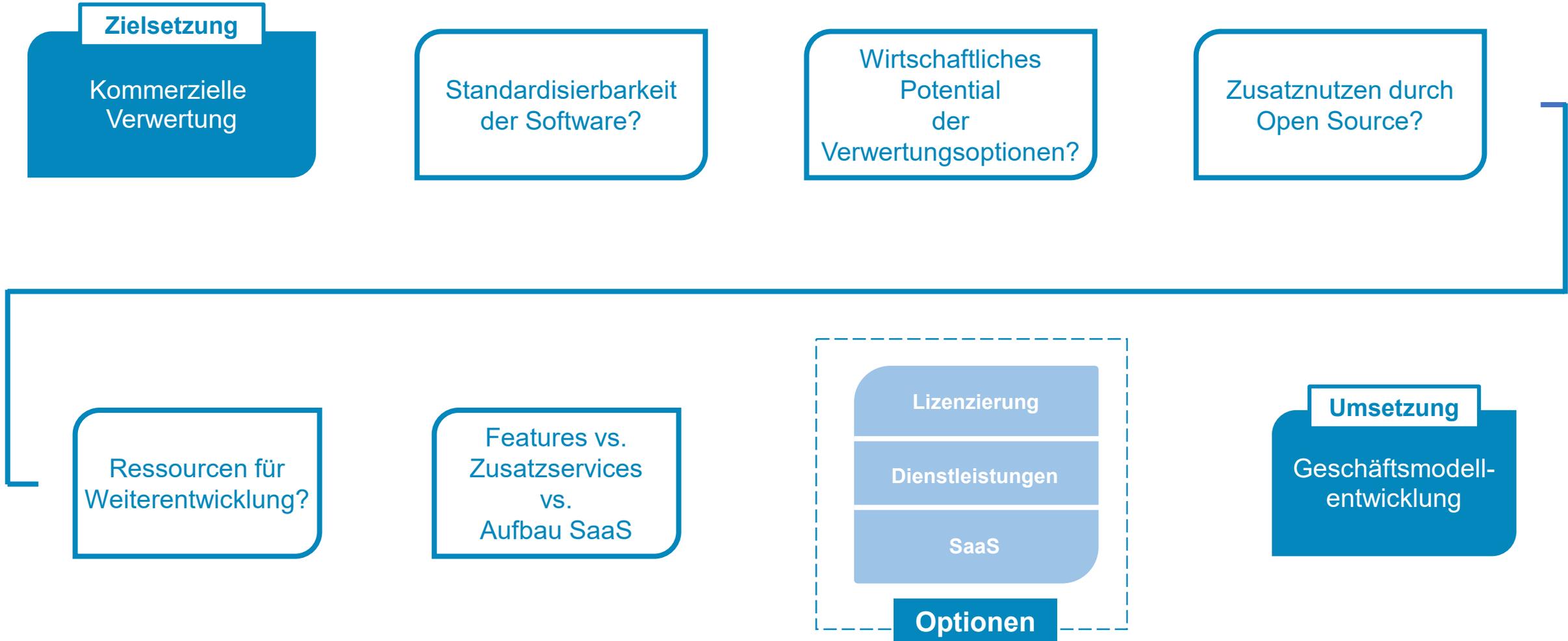


## Lizenzguide

Apache	GPL	AGPL	EUPL
--------	-----	------	------

## Rechtliche Fragestellungen

# Geschäftsentwicklung



# Weiterführende Informationen

## Booklet



SoftWert

Verwertung von Forschungssoftware erfolgreich gestalten

Das Deutsche Elektronen-Synchrotron DESY in Hamburg, das Deutsche GeoForschungsZentrum GFZ in Potsdam, das Deutsche Zentrum für Neurodegenerative Erkrankungen e.V. (DZNE) in Bonn und die Universität des Saarlandes haben gemeinsam einen Methodenbaukasten entwickelt, der Transfermanager:innen Lösungsansätze für die täglichen Herausforderungen im Bereich der Verwertung von Forschungssoftware bieten soll. Dieser umfasst ein Wiki, praxistaugliche Tools und eine Dealdatenbank für Software-Verwertungsfälle.

VIDEO BOOKLET

Hier findet Ihr ausführliche Informationen und Details zu den Erfahrungen, Erkenntnissen, Methoden und Tools. Viel Spaß beim Durchklicken.

Wiki

In diesem Wiki werden die Ergebnisse des Projekts SoftWert vorgestellt. Hier finden sich eine umfassende Informationen zu den verschiedenen Aspekten der Verwertung von Forschungssoftware, aufgeteilt in vier Teilbereiche: Erfassen & Sensibilisieren, Potenzialanalyse, Transferwege und Geschäftsentwicklung. Zusätzlich gibt es hier Hilfestellungen zur Lizenzierung und Ausgründung.

Tools

Ergänzend zu dem Wiki findet sich hier eine vielfältige Sammlung von Tools und Templates, die Transfermanager:innen in ihrer Arbeit unterstützen. Von der Bewertung des Verwertungspotenzials von Forschungssoftware bis zur Auswahl geeigneter Geschäftsmodelle, stehen Werkzeuge zum Download bereit, die jede Phase des Transferprozesses erleichtern.

GoodPractice

Im Projektkonsortium wurden die Methoden nicht nur erarbeitet sondern auch erprobt und teilweise umgesetzt. Die Erfahrungen sind hier in verschiedenen Berichten zusammengefasst.

Ein weiteres Projektergebnis ist die Dealdatenbank, welche erfolgreich abgeschlossene Transferfälle mit

## Wiki



# Agenda

TOP	Thema	Referent
1	Begrüßung und Einleitung	Zahra Saleh
2	Der Methodenbaukasten – die Inhalte	Zahra Saleh
3	<b>Fokusthema 1: Rechtliche Fallstricke</b>	<b>Janine Fischer</b>
<b>15 min Pause</b>		
4	Fokusthema 2: Innovative Transferwege	Lisa Wenzel Michael Jung
5	Good Practice Beispiele aus den Zentren der Projektbeteiligten	Lisa Wenzel Janine Fischer
6	Software Dealdatenbank	Zahra Saleh
7	Take Home Messages	Lisa Wenzel
8	Diskussion und Fragen	alle
<b>Ende des Meetings</b>		

# Rechtliche Fallstricke

---

- Relevanz einer Lizenz
- Urheber- und Verwertungsrecht
- Proprietär versus Open Source
- FOSS Lizenzen
- EULA Lizenzen
- Mythen und Fallstricke

# Rechtliche Fallstricke

---

Darf ich die vorliegende Software nutzen?

Welche der über 100 verschiedene Open Source Lizenzen soll ich wählen?

Darf ich die vorliegende Software weitergeben?

Was ist Copyleft?

Darf ich die vorliegende Software verkaufen oder kommerziell nutzen?

Darf ich die vorliegende Software verändern?

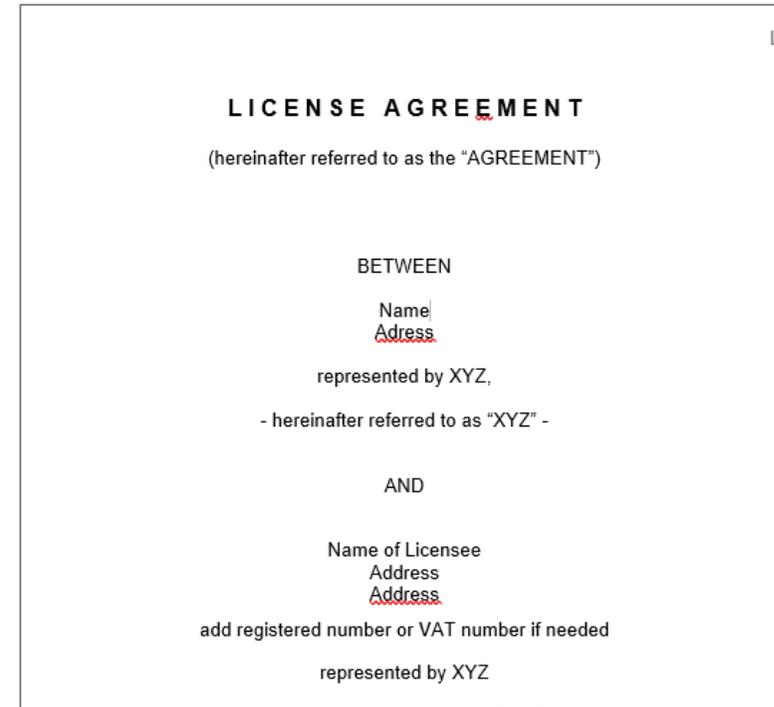
Darf ich die Software einfach mitnehmen, wenn ich meinen Arbeitgeber wechsele?

# Relevanz einer Lizenz



Software für Forschungszwecke oder andere Anwendungen nutzen

➤ Zustimmung der Rechteinhaber erforderlich



**Lizenzen** regeln und legen fest,

- unter welchen Bedingungen
- für welche Zwecke eine Software verwendet werden darf

➤ **Nutzungsbedingungen**



**Lizenzen** werden in der wissenschaftlichen Forschung oft als Hindernis angesehen, da das Thema für Nicht-Juristen:

- **Komplex**
- mit vielen **Herausforderungen** und
- **Mythen** behaftet ist

# Urheberrecht

- Software (Quellcode) ist ein schriftliches Gut
  - Urheberrechtlicher Schutz (**Urheberrechtsgesetz**)
  - automatisch durch die Schaffung des Werkes

## Urheberrechtlich geschützt:

- **Quellcode**, Objektcode, Dokumentation, vorläufige Werke
- von **Menschen** geschaffene Werke

## Urheberrechtlich nicht geschützt:

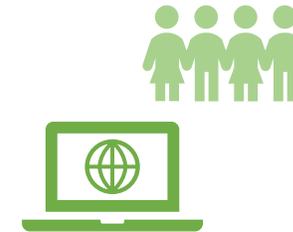
- **Ideen** und Konzepte für Software
- GUIs
- von Algorithmen oder **KIs** geschaffene Werke

- Alle Entwickler der Software, die **aktiv am Code mitgeschrieben** haben, sind Urheber
- Urheberrechte können nicht übertragen, gelöscht oder verkauft werden (deutsches Recht)
- Erlischt 70 Jahre nach dem Tode des letzten Urhebers

# Nutzungs- und Verwertungsrecht

## Gemeinschaftliche Softwareerstellung:

- Alle Urheber haben das ausschließliche Nutzungs- und Verwertungsrecht
- Alle Urheber müssen gemeinschaftlich Verwertungsstrategien zustimmen
- Urheber können auf ihre Verwertungsrechte verzichten / diese übertragen



## Softwareerstellung im Rahmen eines Arbeits- oder Dienstverhältnisses:

- Die ausschließlichen Nutzungs- und Verwertungsrechte liegen beim Arbeitgeber bzw. Dienstherrn
- Bei freien Mitarbeitern verbleiben diese Rechte beim Urheber, es sei denn es ist vertraglich anders geregelt



# Weitere Nutzung festlegen durch Lizenzen

- Insbesondere dann relevant,
  - wenn die Software nicht nur **in-house genutzt**, 
  - sondern auch **kopiert**, 
  - **verändert**, 
  - **nach außen weitergegeben** 
  - oder **kommerziell genutzt** werden soll 
- Lizenzen können all diese Aktivitäten erlauben oder einschränken
- Akademische in-house Nutzung: Lizenz hilfreich → **bei der Weitergabe an Dritte**: Lizenz sinnvoll & notwendig

Die einfache Anwendung einer Software innerhalb einer Firma, ohne Geld damit zu verdienen, ist bereits eine kommerzielle Nutzung

- Lizenzen
  - **selbst erstellen** oder
  - aus bestehenden Vorlagen (z.B **FOSS Lizenzen**) auswählen

# Open Source versus Proprietären Lizenzen

Free Open Source Software (FOSS) Lizenz	Proprietäre Lizenz (End User License Agreement)
<p><b>AGB-ähnlich</b></p> <p><b>standardisiert</b> wie Creative Commons Lizenzen</p> <p>keine Veränderungen am Lizenztext erlaubt</p>	<p><b>Individuell</b> selbst erstellter Lizenztext / Lizenzverträge</p>
<p>Quellcode darf durch den Lizenznehmer</p> <ol style="list-style-type: none"> <li>1. eingesehen werden</li> <li>2. genutzt/angewendet werden</li> <li>3. verändert werden (Modifikation)</li> <li>4. weitergegeben werden</li> </ol>	<p>Alle Nutzungsbedingungen für die Software und den Quellcode können im Lizenztext individuell formuliert und festgelegt werden</p> <p>Quellcode kann, muss aber nicht offengelegt werden</p>
<p>Nutzung der Software muss nicht kostenfrei sein</p>	<p>Nutzung der Software muss nicht Geld kosten</p>

# Free Open Source Software (FOSS) Lizenzen

Auf der Seite der Open Source Initiative sind **über 100 Lizenzen** gelistet  
 (<https://opensource.org/licenses> )

Wahl der Lizenz  
 anhand der  
 eigenen  
 Zielsetzung  
 bzw. Mission

The infographic is set against a dark background with blue icons and text. It is divided into three columns, each with a specific icon and heading.

- Community:** Icon of three people. Text: "I need to work in a community." It recommends using the license preferred by the community and mentions asking maintainers to add a license if a dependency doesn't have one.
- Simple and permissive:** Icon of a double-headed arrow. Text: "I want it simple and permissive." It mentions the MIT License and lists examples like Babel, .NET, and Rails.
- Sharing improvements:** Icon of a refresh/circular arrow. Text: "I care about sharing improvements." It mentions the GNU GPLv3 and lists examples like Ansible, Bash, and GIMP.

Hilfreiche Seite für die Wahl einer Open Source Lizenzen und kurze Darstellung der Bedingungen

Quelle: <https://choosealicense.com/>

# Free Open Source Software (FOSS) Lizenzen

---

- Open Source Lizenzen beinhalten **unterschiedliche Nutzungsbedingungen**
- Lassen sich grob in **drei Gruppen** einteilen:

Starke **Copyleft** Lizenzen  
(GPL 2.0, GPL 3.0, AGPL 3.0, EUPL 1.2)

Schwache **Copyleft** Lizenzen  
(LGPL 2.1, LGPL 3.0, MPL 2.0)

Permissive Lizenzen  
(kein **Copyleft**)  
(MIT, BSD-3, Apache 2.0)

# Copyleft Effekt

---

„Copyleft ist eine allgemeine Methode, ein Programm (oder ein anderes Werk) **frei** (im Sinne von Freiheit, nicht „Nullpreis“) zu machen und zu verlangen, dass alle modifizierten und erweiterten Programmversionen **ebenfalls frei sind**“ (Quelle: [www.gnu.org](http://www.gnu.org)).

Gilt für die:

- **Weitergabe** von Code mit Copyleft Lizenz (sowohl unverändert als auch modifizierter, erweiterter Code)
- **Integration** von Code mit Copyleft Lizenz in den eigenen Code **und Weitergabe** des Gesamtwerks

Copyleft Effekt:

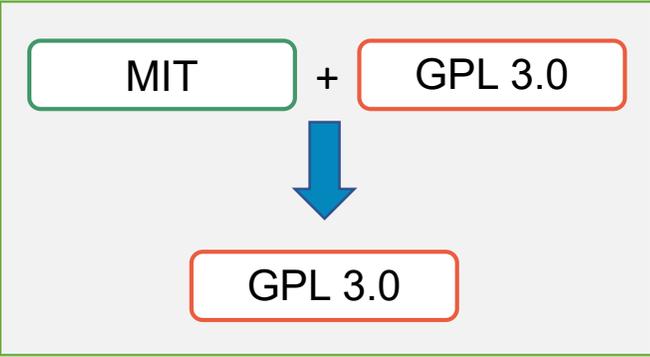
- Copyleft-lizenzierter Code wird in das eigene Software Paket / Code integriert oder statisch verlinkt
  - Die Weitergabe an Dritte nur unter derselben Copyleft-Lizenz
- **kritisch für Firmen** und Entwickler, wenn diese ihren eigenen Code/Software nicht unter eine Copyleft-Lizenz stellen wollen

# Kompatibilität der Open Source Lizenzen

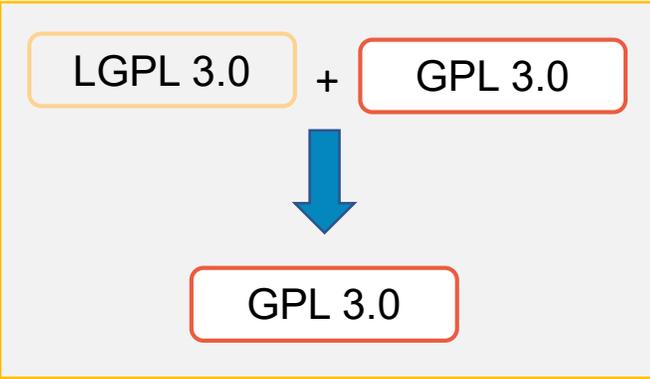
Kann Lizenz A in Lizenz B integriert werden?

A \ B	MIT	BSD3	Apache 2.0	MPL 2.0	LGPL 2.1	LGPL 3.0	GPL 2.0	GPL 3.0	AGPL 3.0	EUPL 1.2
MIT	+	+	+	+	+	+	+	+	+	+
BSD3	- <sup>1</sup>	+	+	+	+	+	+	+	+	+
Apache 2.0	- <sup>2</sup>	- <sup>2</sup>	+	+	+	+	+	+	+	+
MPL 2.0	- <sup>3</sup>	- <sup>3</sup>	- <sup>3</sup>	+	+	+	+	+	+	+
LGPL 2.1	- <sup>3</sup>	- <sup>3</sup>	- <sup>3</sup>	+	+	+	+	+	+	+
LGPL 3.0	- <sup>3</sup>	- <sup>3</sup>	- <sup>3</sup>	- <sup>4</sup>	- <sup>5</sup>	+	- <sup>5</sup>	+	+	+
GPL 2.0	- <sup>3</sup>	- <sup>3</sup>	- <sup>3</sup>	- <sup>6</sup>	- <sup>6</sup>	- <sup>6</sup>	+	- <sup>3</sup>	- <sup>5</sup>	- <sup>7</sup>
GPL 3.0	- <sup>3</sup>	- <sup>3</sup>	- <sup>3</sup>	- <sup>6</sup>	- <sup>6</sup>	- <sup>6</sup>	- <sup>5</sup>	+	+	- <sup>7</sup>
AGPL 3.0	- <sup>3</sup>	- <sup>3</sup>	- <sup>3</sup>	- <sup>6</sup>	- <sup>6</sup>	- <sup>6</sup>	- <sup>5</sup>	- <sup>8</sup>	+	- <sup>7</sup>
EUPL 1.2	- <sup>3</sup>	- <sup>3</sup>	- <sup>3</sup>	- <sup>6</sup>	- <sup>6</sup>	- <sup>6</sup>	+	+	+	+

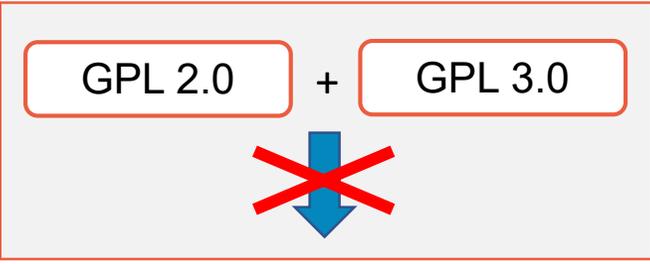
Beispiel 1



Beispiel 2



Beispiel 3



# Open Source Lizenzen

---

- Eigentlich gedacht, um Entwickler dabei zu **unterstützen**, ihre Software **ohne tiefes juristisches Knowhow** an Dritte weiterzugeben oder der Öffentlichkeit zur Verfügung zu stellen (ähnlich CC-BY)
- Open Source Lizenzierung ist allerdings **nicht trivial**

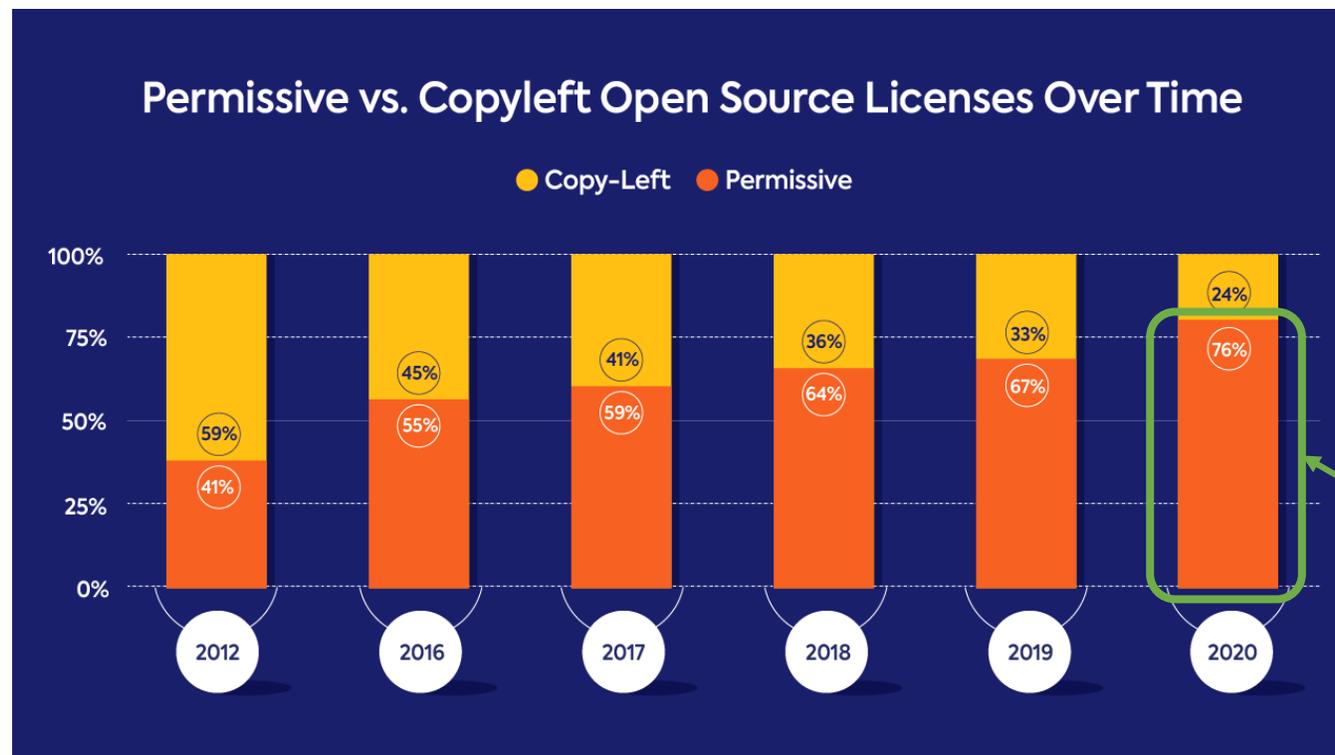
## Lizenzgeber

- Lizenz auswählen anhand der **Rechte**, die für die Weitergabe der Software **gefordert** und **gewährt** werden sollen (z.B. Copyleft)

## Nutzer (Lizenznehmer)

- Wissen, wie der **Quellcode genutzt**, mit anderem Code **verbunden** und **weitergegeben** werden darf (Kompatibilitäten, Copyleft, Urhebernennung, Dokumentation von Änderungen)

# Trend: Permissive Lizenzen

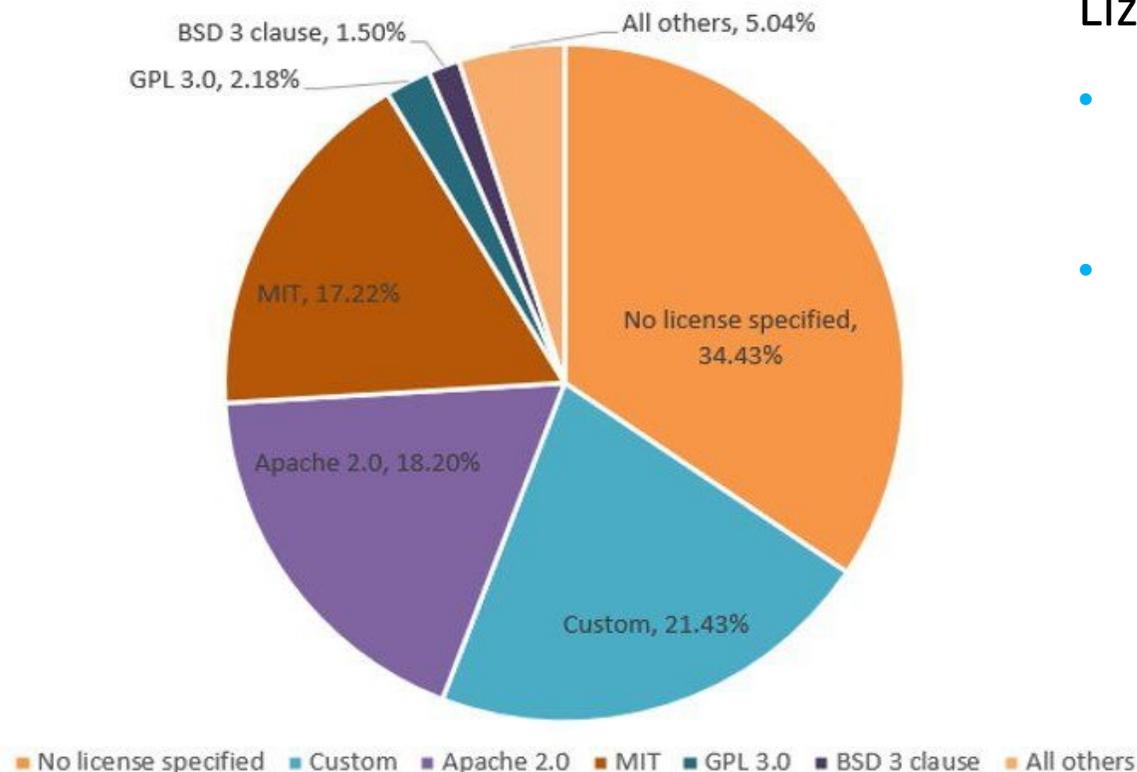


**Permissive Lizenzen  
setzen sich durch**

<https://www.whitesourcesoftware.com/resources/blog/open-source-licenses-trends-and-predictions>

# Trend bei GitHub: FOSS Lizenzen

License Usage on GitHub



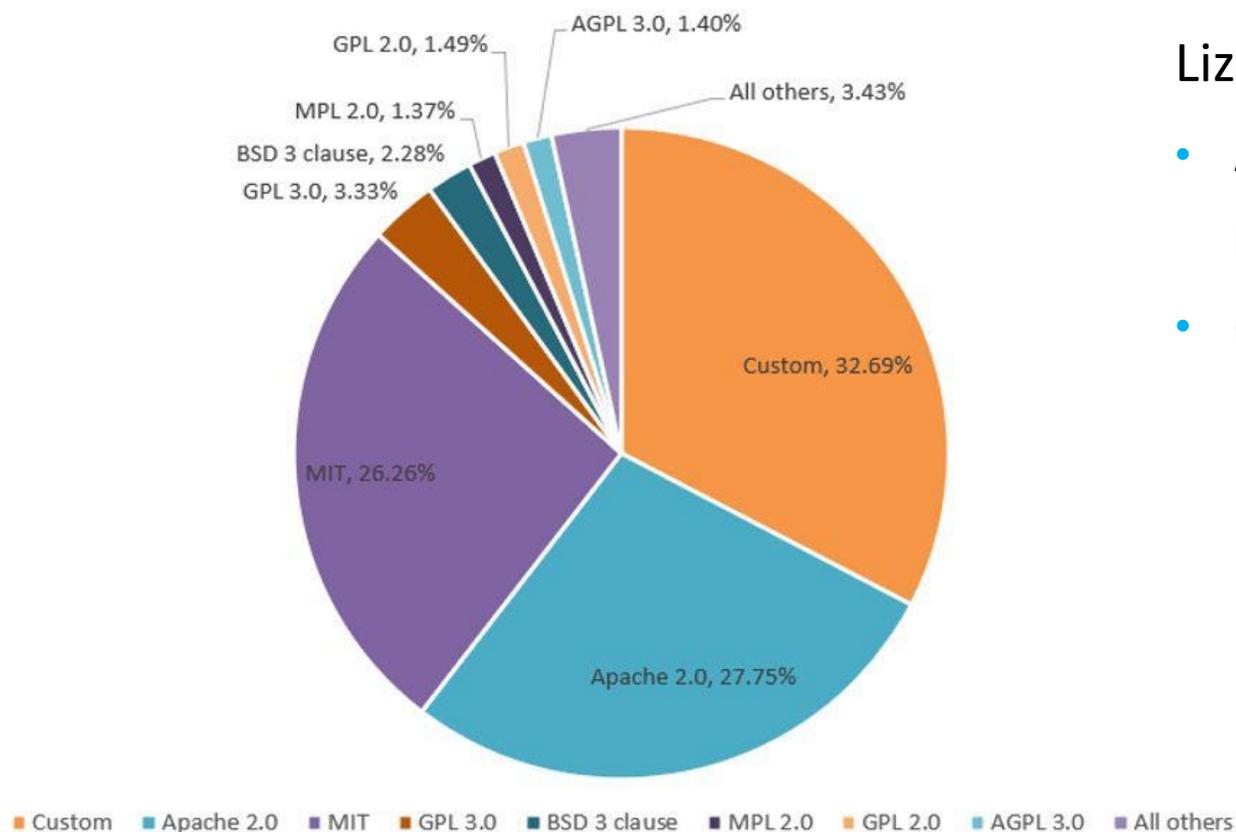
## Lizenzen in Repositorien auf GitHub (2018-2020)

- 34% der GitHub Repositorien: **kein Lizenzfile**
  - Open Source-Status dadurch fraglich
- 21% der GitHub Repositorien: **keine gültige FOSS Lizenz**, sondern custom
  - Durch minimale Veränderungen an einer FOSS Lizenz verliert diese ihre Gültigkeit unter dem jeweiligen Namen und wird zu einer Inviduallizenz, die nicht den Namen der FOSS-Lizenz tragen darf

Data for 2018 - Q2 2020

# Trend bei GitHub: FOSS Lizenzen

License Usage on GitHub Excluding 'No License Specified'



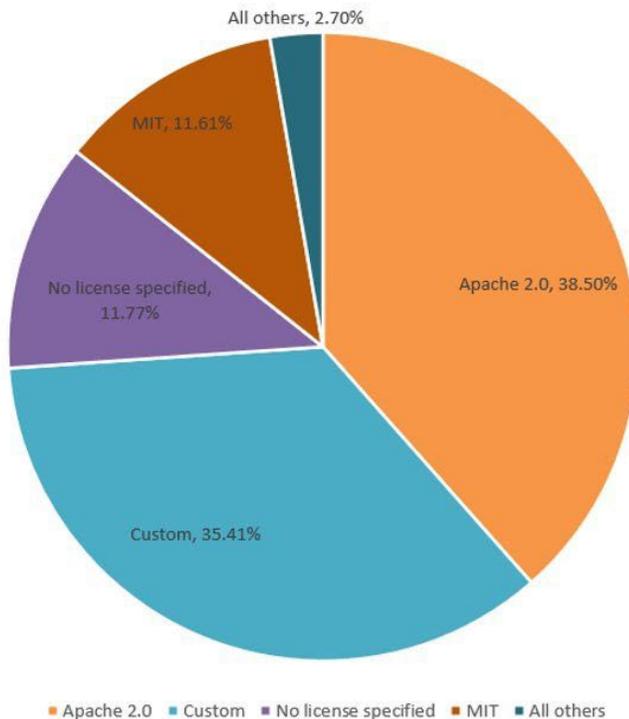
## Lizenzen ohne “no license specified”:

- **Apache 2.0** und **MIT** sind die beiden meist genutzten Lizenzen (54% aller GitHub Repositorien)
- Gefolgt von GPL 3.0 und BSD3

Data for 2018 - Q2 2020

# Trend bei Unternehmen: FOSS Lizenzen

License Usage in Top 5 companies Ranked in OSCI



Data for 2018 - Q1 2020

Lizenznutzung der Top 5 Unternehmen auf GitHub im OSCI Ranking (<https://opensourceindex.io/>):

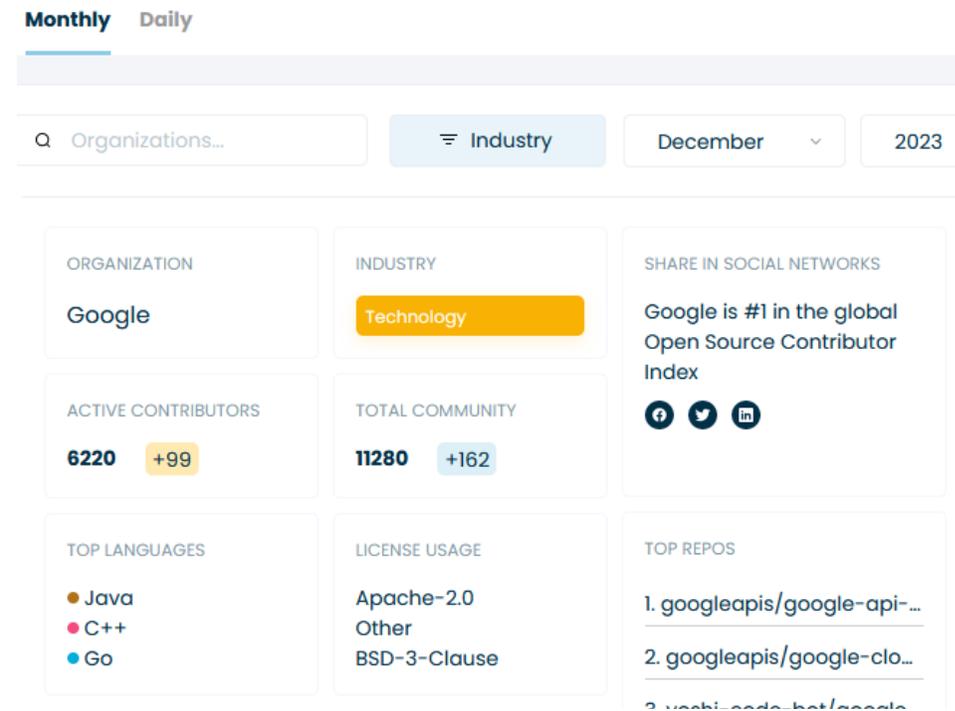
- **Apache 2.0** mit Abstand am meisten genutzt
- **MIT** ist die einzige andere Standardlizenz mit signifikanter Popularität
- Copyleft-Lizenzen wurden kaum genutzt
- Hoher Anteil ohne Lizenzfile; allerdings meist kein Code, sondern größtenteils Beispiele oder Dokumentation

# Trend FOSS Lizenzen

Der Trend bei der Nutzung von Open Source Standardlizenzen (FOSS Lizenzen) geht sowohl bei Unternehmen als auch bei der akademischen Nutzung hin zu

➤ permissiven Lizenzen

1. Apache 2.0
2. MIT
3. BSD3



# Lizenztext – End User License Agreement (EULA)

- Es gibt keine Vorgaben, was in einen Lizenztext gehört
- Der Lizenztext sollte so formuliert sein, dass er sowohl der Lizenzgeber als auch der Lizenznehmer ausreichend rechtlich absichert



**L I C E N S E   A G R E E M E N T**  
(hereinafter referred to as the "AGREEMENT")

BETWEEN

Name  
Address  
represented by XYZ,  
- hereinafter referred to as "XYZ" -

AND

Name of Licensee  
Address  
Address  
add registered number or VAT number if needed  
represented by XYZ  
- hereinafter referred to as "LICENSEE" -

hereinafter jointly referred to as "the Parties", and individually as "the Party" -

conclude the following AGREEMENT:

# Lizenztext – End User License Agreement (EULA)

---

## Mögliche Vertragspunkte in einem Lizenzvertrag:

- Nutzung, Modifikation, Weitergabe (Sublizenzierung)
- Mit Quellcode oder nur ausführbare Software
- Geheimhaltungsklauseln
- Kosten
- Haftung
- Steuerklauseln
- Exportkontrolle
- Gerichtsstand und deutsches Recht



# Mythen im Umgang mit Software

# Mythen im Umgang mit Software

---

Software und Code kann ohne Lizenz verwendet werden

- Keine Software und kein Code darf ohne Erlaubnis verwendet werden
  - Rechte liegen ausschließlich bei den **Rechteinhabern**
- **Rechte** können jedoch **übertragen** werden
  - mittels einer Lizenz
- Eigenem Code, der weitergegeben wird, sollte eine Lizenz beigefügt werden, damit der Nutzer weiß, was er mit dem Code bzw der Software machen darf.

# Mythen im Umgang mit Software

~~Software und Code kann ohne Lizenz verwendet werden~~

- Keine Software und kein Code darf ohne Erlaubnis verwendet werden
  - Rechte liegen ausschließlich bei den **Rechteinhabern**
- **Rechte** können jedoch **übertragen** werden
  - mittels einer Lizenz
- Eigenem Code, der weitergegeben wird, sollte eine Lizenz beigefügt werden, damit der Nutzer weiß, was er mit dem Code bzw der Software machen darf.



# Mythen im Umgang mit Software

Wissenschaftler können ihre Software immer problemlos, einfach und eigenständig Open Source veröffentlichen

Nach **§69b Urheberrechtsgesetz**: „Wird ein Computerprogramm von einem Arbeitnehmer in Wahrnehmung seiner Aufgaben oder nach den Anweisungen seines Arbeitgebers geschaffen, so ist **ausschließlich der Arbeitgeber zur Ausübung aller vermögensrechtlichen Befugnisse an dem Computerprogramm berechtigt, sofern nichts anderes vereinbart ist.**“

- ausschließliche Verwertungsrechte gehen in dem Fall auf den Arbeitgeber über
- Urheberrechte verbleiben bei den Urhebern des Codes

In Arbeitsverhältnissen gibt es **Regelungen oder Policies**, die festlegen, wer die Erlaubnis zur Open Source Lizenzierung erteilen darf, z.B. Vorgesetzte oder spezielle Ansprechpartner

**Vorgaben** durch Drittmittelgeber oder Policies in Konsortien (z.B. CTAO)

# Mythen im Umgang mit Software

Wissenschaftler können ihre Software immer problemlos, einfach und eigenständig Open Source veröffentlichen

Nach **§69b Urheberrechtsgesetz**: „Wird ein Computerprogramm von einem Arbeitnehmer in Wahrnehmung seiner Aufgaben oder nach den Anweisungen seines Arbeitgebers geschaffen, so ist **ausschließlich der Arbeitgeber zur Ausübung aller vermögensrechtlichen Befugnisse an dem Computerprogramm berechtigt, sofern nichts anderes vereinbart ist.**“

- ausschließliche Verwertungsrechte gehen in dem Fall auf den Arbeitgeber über
- Urheberrechte verbleiben bei den Urhebern des Codes

In Arbeitsverhältnissen gibt es **Regelungen oder Policies**, die festlegen, wer die Erlaubnis zur Open Source Lizenzierung erteilen darf, z.B. Vorgesetzte oder spezielle Ansprechpartner

**Vorgaben** durch Drittmittelgeber oder Policies in Konsortien (z.B. CTAO)



# Mythen im Umgang mit Software

---

Alle Open Source Lizenzen sind gleich. Es ist egal welche ich nehme.

- Die verschiedene FOSS Lizenzen legen **unterschiedliche Nutzungsbedingungen** fest.
- Beim Zusammenführen unterschiedlich lizenzierter Software kann es zu Inkompatibilitäten kommen.
- Insbesondere eine **Kompatibilitätsprüfung** ist daher unerlässlich.

# Mythen im Umgang mit Software

Alle Open Source Lizenzen sind gleich. Es ist egal welche ich nehme.

- Die verschiedene FOSS Lizenzen legen **unterschiedliche Nutzungsbedingungen** fest.
- Beim Zusammenführen unterschiedlich lizenzierter Software kann es zu Inkompatibilitäten kommen.
- Insbesondere eine **Kompatibilitätsprüfung** ist daher unerlässlich.



# Mythen im Umgang mit Software

Statt FOSS Lizenzen können genauso gut **Creative Commons** (z.B. CC-BY) Lizenzen verwendet werden.

Es gibt da keine Unterschiede

- Software Code ist auch nur ein Text, welcher in einer bestimmen (Programmier-)Sprache verfasst ist
- Allerdings wird aber davon **abgeraten** CC-Lizenzen für die Lizenzierung von Software zu verwenden, da in den CC-Lizenzen beispielsweise **folgende Punkte fehlen**:
  - die Bedingungen für Offenlegung des Quellcodes
  - klare Anweisungen für Modifikationen von Software
- Besser eine auf Software ausgelegte Lizenz verwendet werden, wie zum Beispiel die FOSS Lizenzen.

# Mythen im Umgang mit Software

Statt FOSS Lizenzen können genauso gut **Creative Commons** (z.B. CC-BY) Lizenzen verwendet werden.

Es gibt da keine Unterschiede

- Software Code ist auch nur ein Text, welcher in einer bestimmen (Programmier-)Sprache verfasst ist
- Allerdings wird aber davon **abgeraten** CC-Lizenzen für die Lizenzierung von Software zu verwenden, da in den CC-Lizenzen beispielsweise **folgende Punkte fehlen**:
  - die Bedingungen für Offenlegung des Quellcodes
  - klare Anweisungen für Modifikationen von Software
- Besser eine auf Software ausgelegte Lizenz verwendet werden, wie zum Beispiel die FOSS Lizenzen.



# Mythen im Umgang mit Software

---

Open Source bedeutet kostenlos

Mit Open Source lizenzierter Software kann man kein Geld verdienen

**Open Source bedeutet nicht kostenlos, sondern dass der Quellcode (Source) offengelegt wird**

Für eine Open Source veröffentlichte Software kann Geld genommen werden.

Lediglich für die Open Source Lizenz an sich darf kein Geld verlangt werden.

# Mythen im Umgang mit Software

**Open Source bedeutet kostenlos**

**Mit Open Source lizenzierter Software kann man kein Geld verdienen**

**Open Source bedeutet nicht kostenlos, sondern dass der Quellcode (Source) offengelegt wird**

Für eine Open Source veröffentlichte Software kann Geld genommen werden.

Lediglich für die Open Source Lizenz an sich darf kein Geld verlangt werden.



# Mythen im Umgang mit Software

---

Ist eine Software erst einmal Open Source lizenziert, ist eine kommerzielle Nutzung nicht mehr möglich

Wenn die Verwertungsrechte alle innerhalb der eigenen Institution (keine Miturheber mit anderen Arbeitgebern) und keine weiteren Einschränkungen bestehen,

- können auch weitere proprietäre Lizenzen vergeben werden
- ist eine kommerzielle Verwertung z.B. durch **Dual Licensing** durchaus möglich.

Darüber hinaus kann Open Source Software durch **SaaS, Dienstleistungen, Beratung** etc. weitere Einnahmen generieren.

# Mythen im Umgang mit Software

Ist eine Software erst einmal Open Source lizenziert, ist eine kommerzielle Nutzung nicht mehr möglich

Wenn die Verwertungsrechte alle innerhalb der eigenen Institution (keine Miturheber mit anderen Arbeitgebern) und keine weiteren Einschränkungen bestehen,

- können auch weitere proprietäre Lizenzen vergeben werden
- ist eine kommerzielle Verwertung z.B. durch **Dual Licensing** durchaus möglich.



Darüber hinaus kann Open Source Software durch **SaaS, Dienstleistungen, Beratung** etc. weitere Einnahmen generieren.

# Mythen im Umgang mit Software

---

Ich kann meine Software bei einem **Jobwechsel** einfach mitnehmen und im neuen Anstellungsverhältnis weiternutzen

- Wenn die Software im Rahmen des Arbeitsvertrages erstellt wurde, **verfügt der alte Arbeitgeber über die Verwertungsrechte**
- Die Mitnahme und Weiternutzung/Weiterentwicklung der Software beim neuen Arbeitgeber stellt eine **Weitergabe an Dritte** dar
- Eine Möglichkeit der weiteren Nutzung der Software kann zwischen dem alten und dem neuen Arbeitgeber über eine **Lizenz** geregelt werden

# Mythen im Umgang mit Software

Ich kann meine Software bei einem **Jobwechsel** einfach mitnehmen und im neuen Anstellungsverhältnis weiternutzen

- Wenn die Software im Rahmen des Arbeitsvertrages erstellt wurde, **verfügt der alte Arbeitgeber über die Verwertungsrechte**
- Die Mitnahme und Weiternutzung/Weiterentwicklung der Software beim neuen Arbeitgeber stellt eine **Weitergabe an Dritte** dar
- Eine Möglichkeit der weiteren Nutzung der Software kann zwischen dem alten und dem neuen Arbeitgeber über eine **Lizenz** geregelt werden



# Mythen im Umgang mit Software

Studenten **ohne Arbeitsvertrag** können problemlos **Code zur Software beitragen**.

Codebeiträge von einem guten Kollegen aus einer **anderen Forschungseinrichtung** kann ich einfach in meinen Code einfügen und alleine die Lizenz festlegen

- Jede Person, die aktiv Code schreibt und zur Software beiträgt, ist automatisch **Miturheber**
- Alle Miturheber der Software stimmen **gemeinsam** über die Lizenzwahl und die Verwertung ab
- Bei Miturhebern, die einen Arbeitsvertrag mit einer **anderen** Forschungseinrichtung / Universität haben, müssen die Verwertungsrechte mit den **Arbeitgebern** geklärt werden
- Für Personen ohne Arbeitsvertrag bei der eigenen Institution
  - Rechte z.B. mittels einem **Contributor License Agreement (CLA)** regeln oder übertragen
  - sind Regelungen oder Rechteübertragung auch für Beteiligung an Open Source Projekten sinnvoll

# Mythen im Umgang mit Software

Studenten **ohne Arbeitsvertrag** können problemlos **Code zur Software beitragen**.

Codebeiträge von einem guten Kollegen aus einer **anderen Forschungseinrichtung** kann ich einfach in meinen Code einfügen und alleine die Lizenz festlegen

- Jede Person, die aktiv Code schreibt und zur Software beiträgt, ist automatisch **Miturheber**
- Alle Miturheber der Software stimmen **gemeinsam** über die Lizenzwahl und die Verwertung ab
- Bei Miturhebern, die einen Arbeitsvertrag mit einer **anderen** Forschungseinrichtung / Universität haben, müssen die Verwertungsrechte mit den **Arbeitgebern** geklärt werden
- Für Personen ohne Arbeitsvertrag bei der eigenen Institution
  - Rechte z.B. mittels einem **Contributor License Agreement (CLA)** regeln oder übertragen
  - sind Regelungen oder Rechteübertragung auch für Beteiligung an Open Source Projekten sinnvoll



# Mythen im Umgang mit Software

---

## Copyright-Vermerk = Urheberrechtsvermerk

### Der Copyright-Vermerk

- hat **nichts** mit den **Urheberpersönlichkeitsrechten** der Autoren zu tun (diese verbleiben nach deutschem Gesetz bei den Urhebern)
- gibt an, wer über die **Verwertungsrechte** verfügt, sei es der Arbeitgeber oder der Autor selbst. Dies kann nach § 69b UrhG auch der Arbeitgeber sein. Ansonsten sind es die Urheber.

# Mythen im Umgang mit Software

Copyright-Vermerk = Urheberrechtsvermerk

## Der Copyright-Vermerk

- hat **nichts** mit den **Urheberpersönlichkeitsrechten** der Autoren zu tun (diese verbleiben nach deutschem Gesetz bei den Urhebern)
- gibt an, wer über die **Verwertungsrechte** verfügt, sei es der Arbeitgeber oder der Autor selbst. Dies kann nach § 69b UrhG auch der Arbeitgeber sein. Ansonsten sind es die Urheber.



# Good Practise: Copyright-Vermerk im code header

```
<one line to give the program's name and a brief idea of what it does.>
```

```
Copyright (C) <year> Deutsches Elektronensynchrotron DESY
```

Inhaber der Verwertungsrechte

```
- <name of author>, <name of employer> <year>
```

Inhaber der Urheberpersönlichkeitsrechte

```
- <name of author>, <name of employer> <year>
```

```
- ...
```

<https://spdx.dev/learn/handling-license-info/>

```
// SPDX-License-Identifier: Apache-2.0
```

```
Licensed under the Apache License, Version 2.0 (the "License");
```

```
you may not use this file except in compliance with the License.
```

```
You may obtain a copy of the License at
```

```
http://www.apache.org/licenses/LICENSE-2.0
```

```
Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.
```

# Agenda

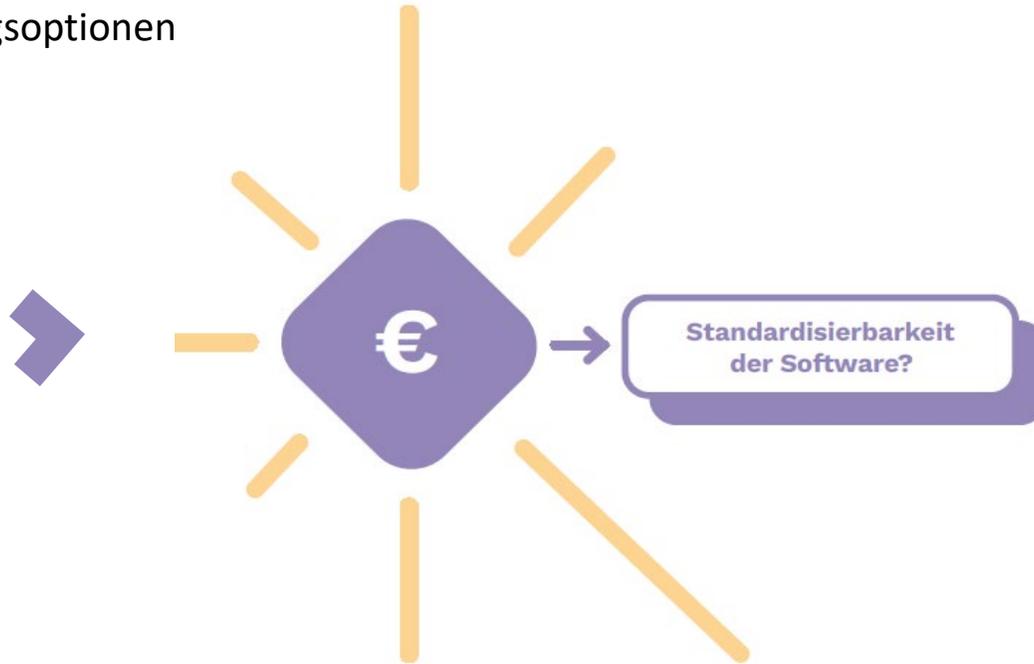
TOP	Thema	Referent
1	Begrüßung und Einleitung	Zahra Saleh
2	Der Methodenbaukasten – die Inhalte	Zahra Saleh
3	Fokusthema 1: Rechtliche Fallstricke	Janine Fischer
<b>15 min Pause</b>		
4	<b>Fokusthema 2: Innovative Transferwege</b>	<b>Lisa Wenzel Michael Jung</b>
5	Good Practice Beispiele aus den Zentren der Projektbeteiligten	Lisa Wenzel Janine Fischer
6	Software Dealdatenbank	Zahra Saleh
7	Take Home Messages	Lisa Wenzel
8	Diskussion und Fragen	alle
<b>Ende des Meetings</b>		

# Innovative Transferwege - Geschäftsentwicklung

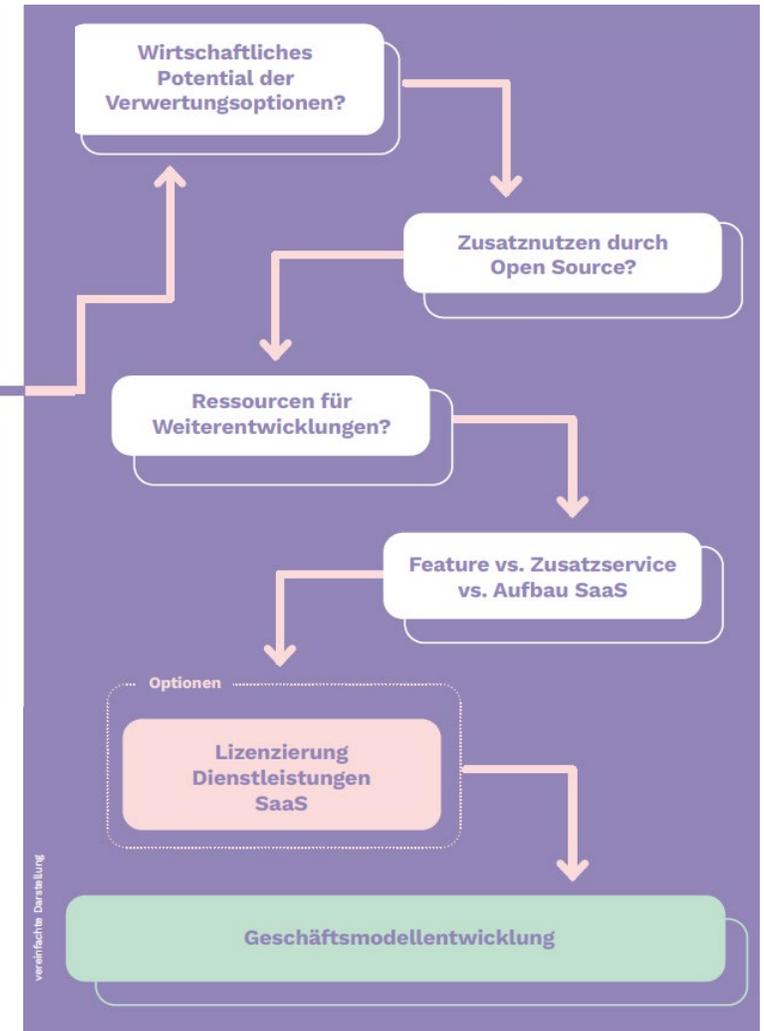
Ein **Entscheidungsbaum** als Grundlage für die Auswahl und Kombination verschiedener Verwertungsoptionen



Mögliche Transferwege



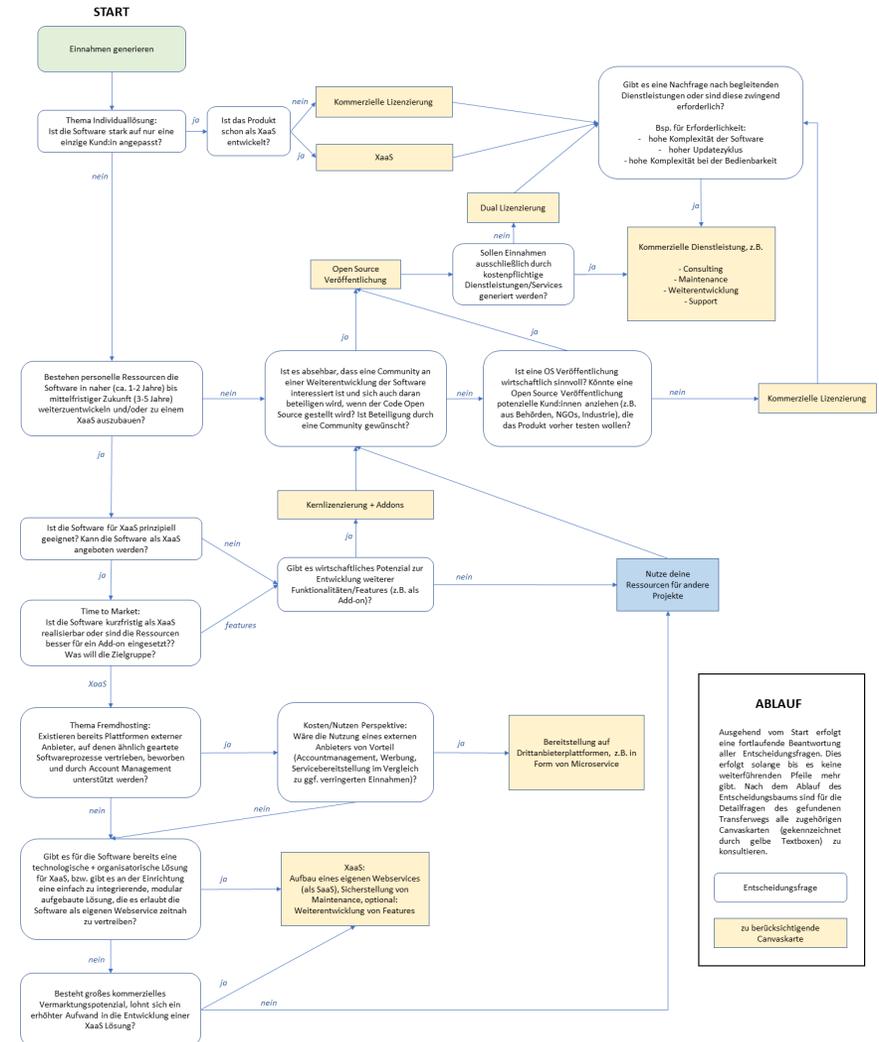
- Während des Durchlaufs des Entscheidungsbaums ist es möglich, mehrere Verwertungsoptionen auszuwählen und so miteinander zu kombinieren
- Je nachdem, welches Set an Verwertungsoptionen ausgewählt wurde, können für eine zielgerichtete Geschäftsmodellentwicklung zusätzliche Fragestellungen beantwortet werden



# Innovative Transferwege – Entscheidungsbaum

## Ausgangsbasis:

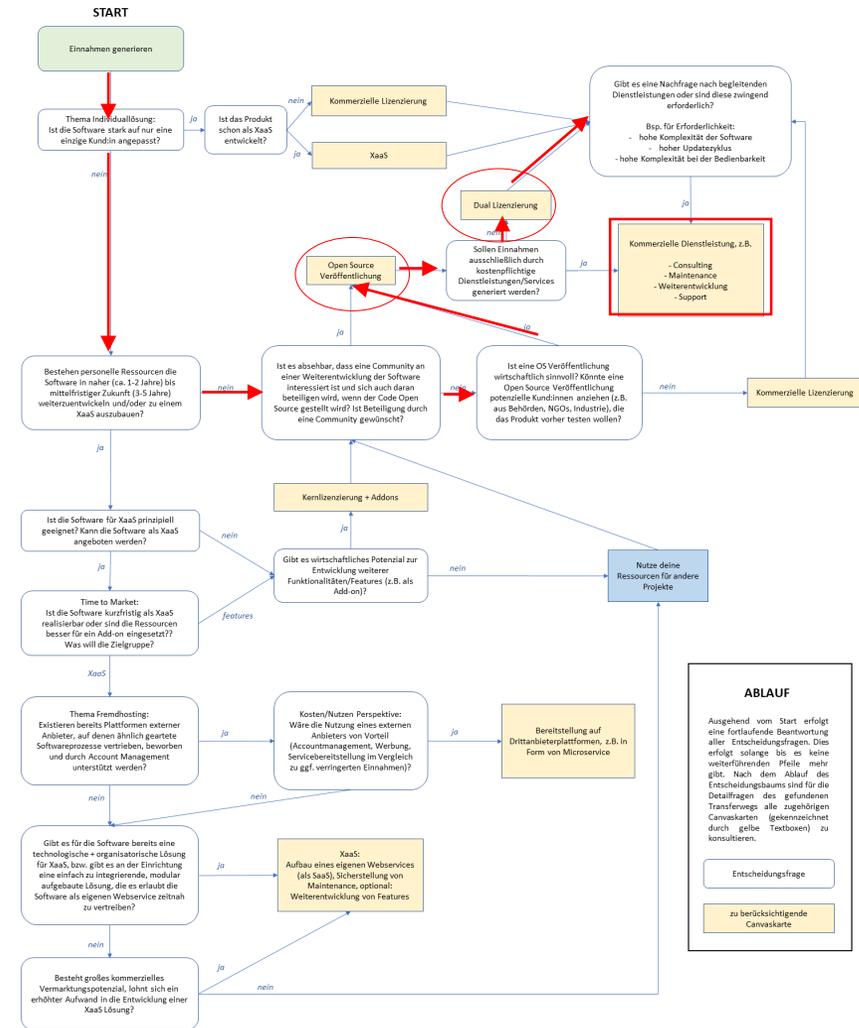
- Forschungssoftware im Bereich der Fernerkundung
- kleines Entwicklerteam (<5 Personen), von denen alle beim GFZ angestellt sind
- ein Interesse intern und für die Wissenschaft, aber auch Potential für außerwissenschaftliche Zielgruppen (NGOs, Behörden, Industrie) vorhanden
- Keine Lizenz-Restriktionen
- Einnahmen sollen generiert werden



# Innovative Transferwege – Entscheidungsbaum

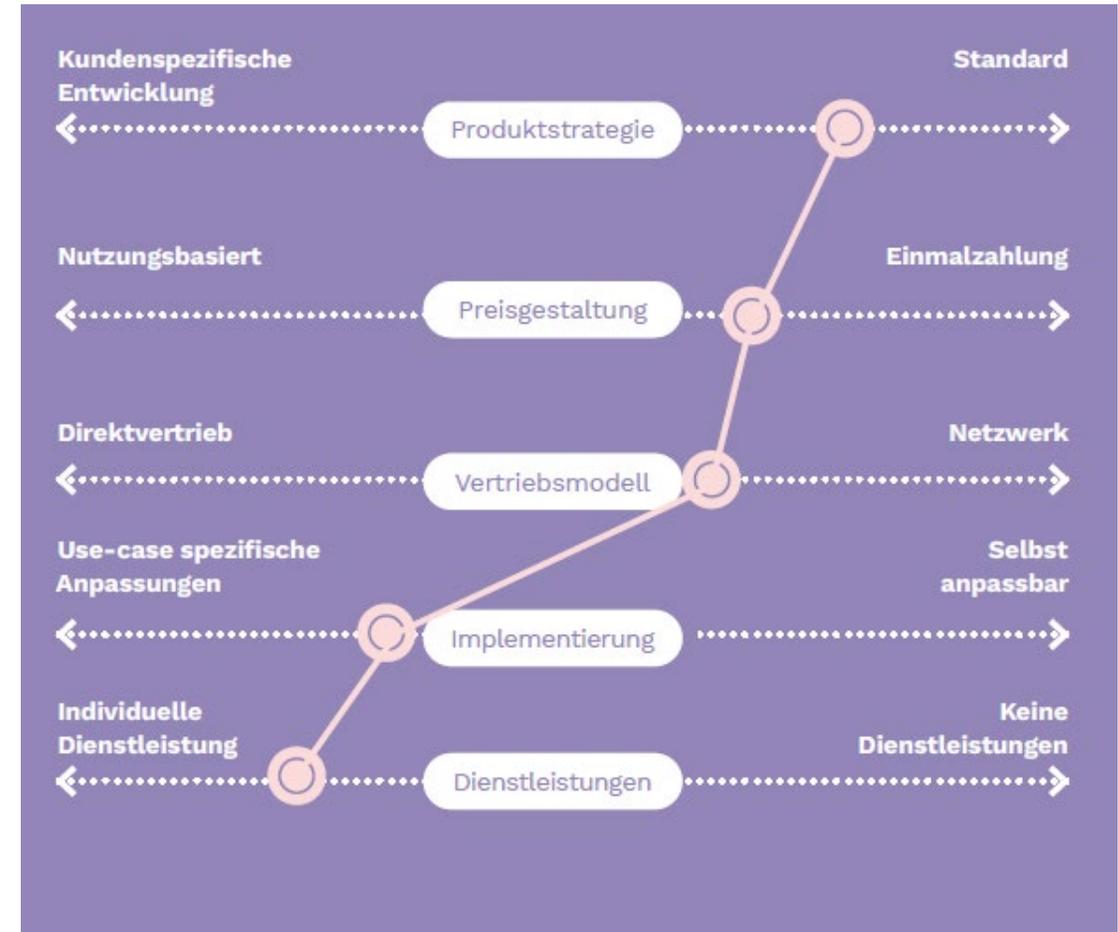
## Fragen/Weg durch den Entscheidungsbaum:

- Ist die Software stark auf nur einen oder eine einzige Kund:in angepasst? → **nein**
- Bestehen personelle Ressourcen die Software in naher oder mittelfristiger Zukunft weiterzuentwickeln und/oder zu einem SaaS auszubauen? → **nein**
- Ist es absehbar, dass eine Community an der Weiterentwicklung interessiert ist und sich auch daran beteiligen wird, wenn der Code Open Source gestellt wird? Ist eine Beteiligung durch die Community gewünscht? → **nein**
- Ist eine OS Veröffentlichung wirtschaftlich sinnvoll? Könnte eine Open Source Veröffentlichung potenzielle Kund:innen anziehen (z.B. aus Behörden, NGOs, Industrie), die das Produkt vorher testen wollen? → **ja** → Geschäftsoption Open Source Veröffentlichung → Pfeil weiter verfolgen
- Sollen Einnahmen ausschließlich durch kostenpflichtige Dienstleistungen/Services generiert werden? → **nein** (Lizenzeeinnahmen gewünscht) → Geschäftsfeld Dual Lizenzierung → Pfeil weiter verfolgen
- Gibt es eine Nachfrage nach begleitenden Dienstleistungen oder sind diese zwingend erforderlich? → **zu evaluieren**



# Innovative Transferwege – Geschäftsmodellentwicklung

- Kommerzielle Lizenzierung, Open Source  
Veröffentlichung, Kernlizenzierung + Add-ons, SaaS oder Bereitstellung auf Drittanbieterplattformen, Dual Lizenzierung, Kommerzielle Dienstleistungen
- Hilfestellungen zur **Geschäftsmodellentwicklung** für bestimmte Verwertungswege bzw. –optionen befinden sich im **SoftWert Wiki**
- Für jede Option werden dann Ausprägungen bzw. zu beachtenden Fragestellungen für **verschiedene Dimensionen** der Geschäftsmodellentwicklung mitgegeben



# Innovative Transferwege – Open Science & Transfer

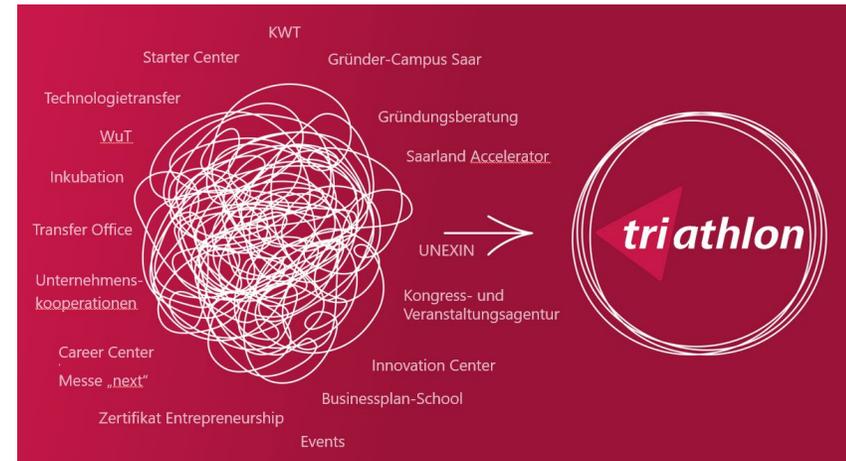
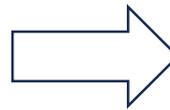
---

## Fragen/Weg durch den Entscheidungsbaum:

- **Dual-Licensing** bietet sich als Transferweg und Geschäftsmodell-Option für den hier beschriebenen Fall an, da sowohl Bedarfe für die wissenschaftliche Community als auch für wirtschaftliche Stakeholder identifiziert wurden
- Die Software wird zunächst erst einmal nur als Open-Source Variante zur Verfügung gestellt und evaluiert, ob die **Software eine Nachfrage bedienen kann und ob zusätzliche Features erwünscht sind** (von wirtschaftlichen Stakeholdern, die an Entwicklerteam heran treten)
- Software wird beworben, z. B. Helmholtz Research Directory, Zenodo, PyPi, Teamwebsite, LinkedIn
- Für die Open-Source Variante wird eine Lizenz mit **strengem Copyleft** gewählt → EUPL v1.2
- Spätere proprietäre Variante: **EULA** (Weiterentwicklung/Updates/Wartung)

# Der Projektpartner Universität des Saarlandes

	<p><b>Seit 1948</b> (Deutsch-französische Gründung)</p>	<p><b>17.000</b> Studierende (30% International)</p>
<p>Campus Saarbrücken &amp; Homburg</p>	<p><b>Sechs Fakultäten:</b> Empirische Humanwissenschaften und Wirtschaftswissenschaft   Medizin   Mathematik und Informatik   Naturwissenschaftlich-Technische   Philosophische   Rechtswissenschaftliche Fakultät</p>	
<p><b>Führende Gründer-Hochschule</b></p>	<p>Sieben <b>Graduierten-Kollegs</b></p>	<p>Sieben <b>Sonder-Forschungsbereiche</b></p>

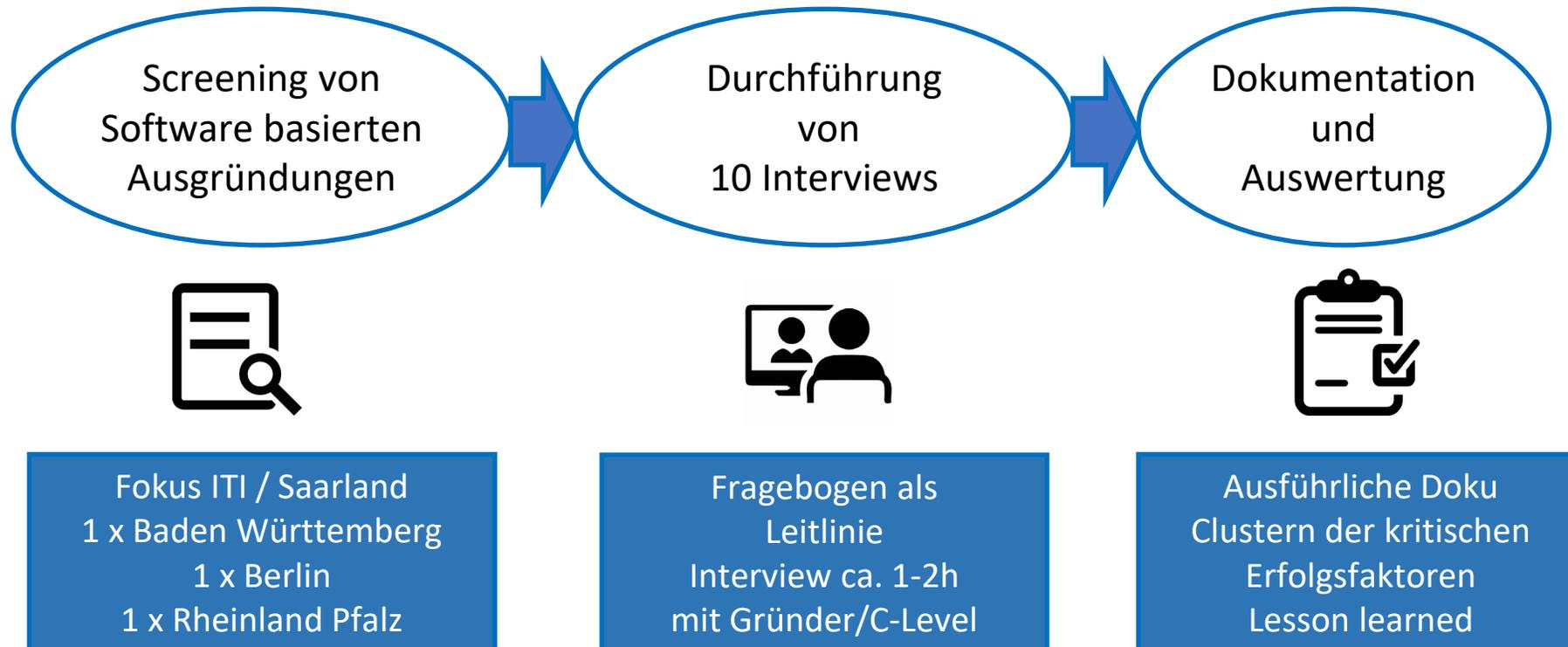


**Bündelung aller Aktivitäten in Triathlon**  
Das Ökosystem für Entrepreneurship, Innovation und Transfer

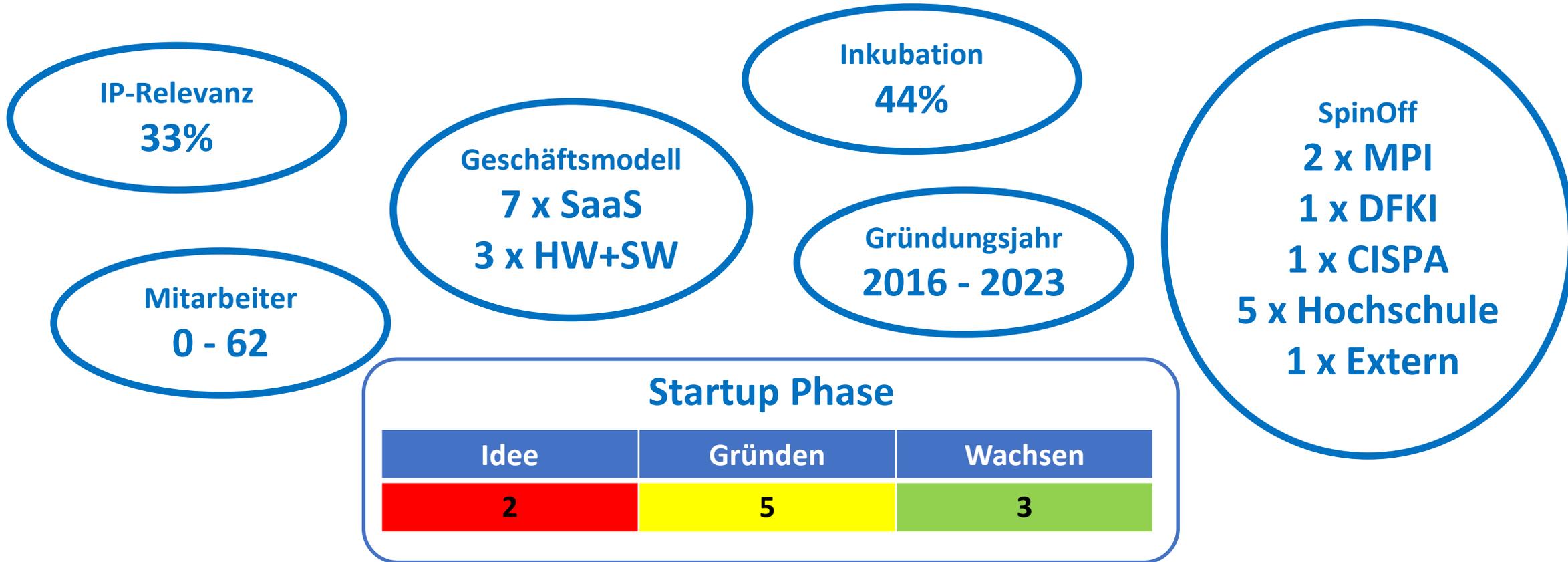


# Ausgründung

## Fokus - Best Practice Transfer mit Software



# Analyse der Interviews



Idee	Gründen	Wachsen

# Kurzdarstellung Interviews – Case 1

Background: Team aus UdS-Lehrstuhl entwickelt intelligenten Algorithmus zur Batterie-Optimierung von LEO-Satelliten (Low Earth Orbit – Satelliten)

Geschäftsmodell	IP-Relevanz	Lizenzsituation	TRL / POC
geplant als SaaS / B2B kostenlose Nutzung für die Wissenschaft 	Kritisch, da ERC Veröffentlichungen (Paper) fördert 	Erfindungsmeldung offen Klärung der Lizenz- bedingungen 	2 Jahre Entwicklungszeit PoC wurde erbracht 
Team	Finanzierung	Support	Kritische Erfolgsfaktoren
Team aus UdS-Lehrstuhl 	ERC Grant ERC PoC Grant 	IT-Inkubator 0,5 FTE als Projekt-Unterstützung 	Team Setup Anschlussfinanzierung Veröffentlichung unklare Vermarktungs- situation 

Idee	Gründen	Wachsen

# Kurzdarstellung Interviews – Case 2

Background: Team aus ext. Institut (CISPA) – IP von Institut ist für Team frei nutzbar – Bereich Automotive – Gründung in 2021

Geschäftsmodell	IP-Relevanz	Lizenzsituation	TRL / POC
B2B / SaaS Individuelles modulares Automobil Customizing 	Alt-IP aus Institut frei Neu-IP aus Inkubation übertragen (Beteilig.) Exist-IP (Uni) 	Lizenzvertrag mit Uni IP wird verkauft 	Kein MVP TRL zu Beginn niedrig 
Team	Finanzierung	Support	Kritische Erfolgsfaktoren
Mixed-Team Schwerpunkt Tech 	Inkubator Exist FT Seed über VC + Strategie 	Inkubator Infrastruktur (Raum) kritisch 	Kein Pilotkunde Fokus Entwicklung Networking F&E Automotive 

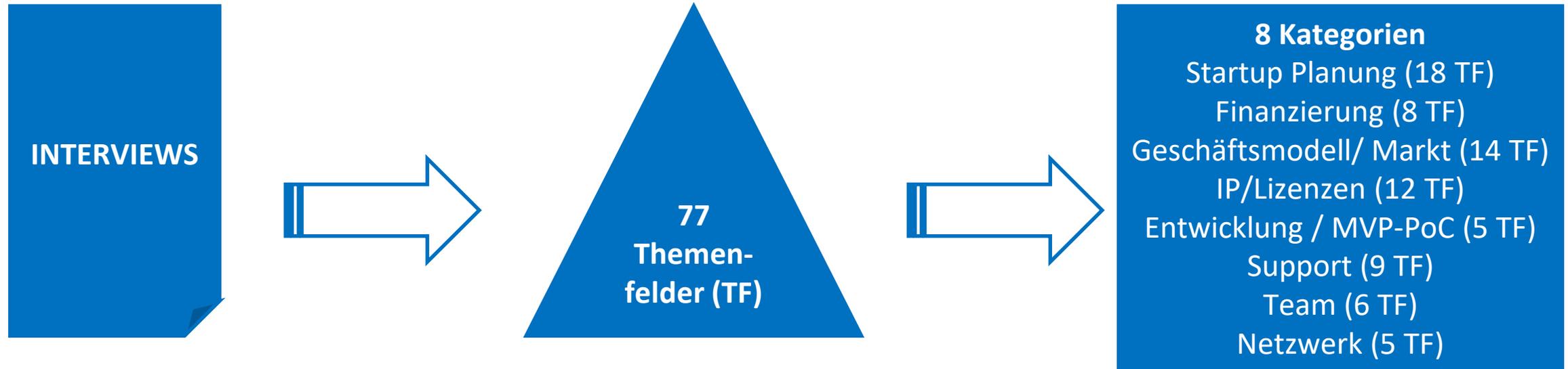
Idee	Gründen	Wachsen

# Kurzdarstellung Interviews – Case 3

Background: Postdoc am Exzellenz Cluster MPI – Innovation in Forschungsgruppe HW/SW entstanden  
 2016 Gründung - 2018 erste Finanzierung - 2020 erste Umsätze

Geschäftsmodell	IP-Relevanz	Lizenzsituation	TRL / POC
B2C / B2B Hardware mit Softwarekomponenten 	Patentanm. von MPI + Uni vor Gründung Übertragung anschließend zus. Anteile 	Lizenzvertrag vereinb. Code frei von Rechten Dritten 	Kein MVP TRL zu Beginn niedrig 
Team	Finanzierung	Support	Kritische Erfolgsfaktoren
Ideengeber nicht im Gründungsteam Mix-Team externer CEO 	Preseed Seed kein durchgängiges Fundraising 	Inkubation mehr Unterstützung in rechtlichen und finanziellen Themen 	Reifegrad zu Beginn IP Klärung Partnerschaften 

# Kritische Erfolgsfaktoren



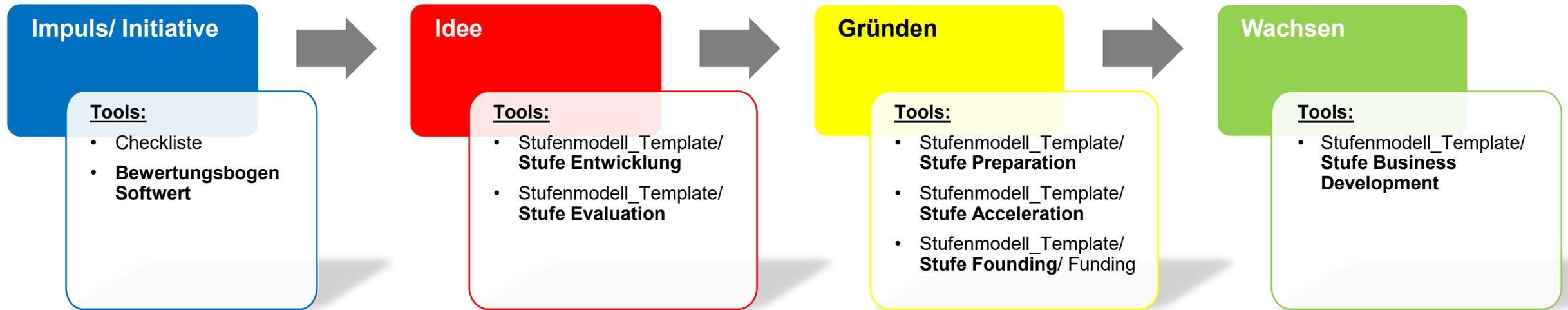
Auf Basis der Interviews wurden insgesamt **77 Themenfelder** analysiert die wiederum in **8 Kategorien** zusammengefasst wurden  
 => **Kritische Erfolgsfaktoren (KEF)**

# Kritische Erfolgsfaktoren

Die **weitere Analyse der KEF** auf Basis der Nennungen pro Themenfeld über alle Interviews (n=10) kam zu folgendem **Trend**, bzw. Ranking:

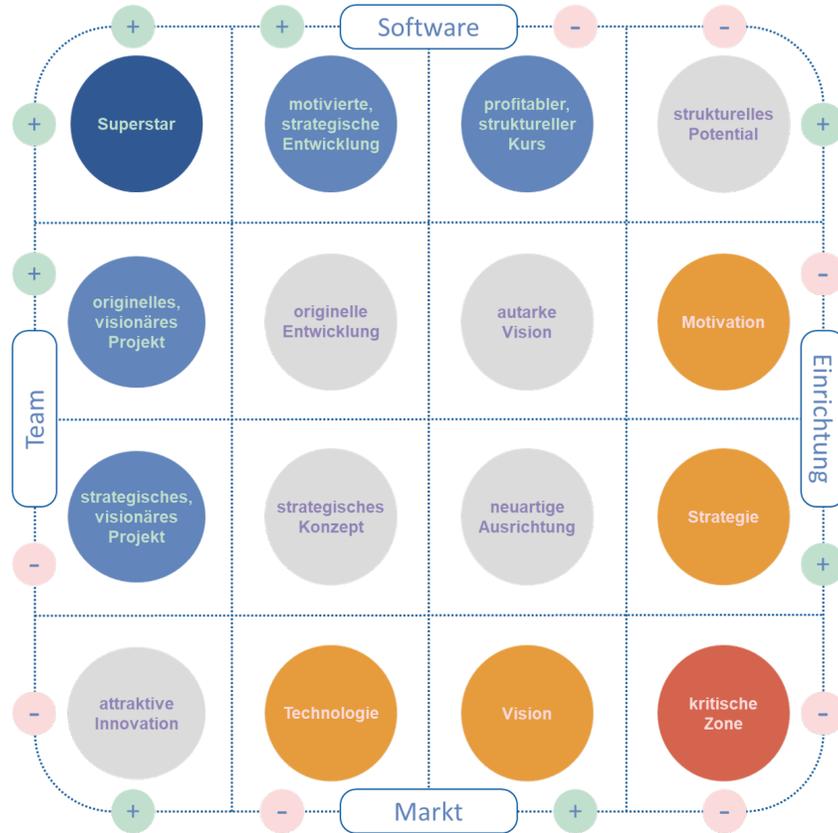
	Kritische Erfolgsfaktoren	Themenfelder	Nennung	Faktor
1	Entwicklung / MVP – PoC	5	19	<b>3,8</b>
2	Finanzierung	8	26	<b>3,25</b>
3	Team	6	17	<b>2,83</b>
4	IP / Lizenzen	12	24	<b>2,00</b>
5	Netzwerk	5	9	<b>1,80</b>
6	Startup-Planung	18	27	<b>1,50</b>
7	Geschäftsmodell / Markt	14	21	<b>1,50</b>
8	Support	9	13	<b>1,44</b>

# Gründungsprozess Tool-Box



# Bewertung des Verwertungspotentials

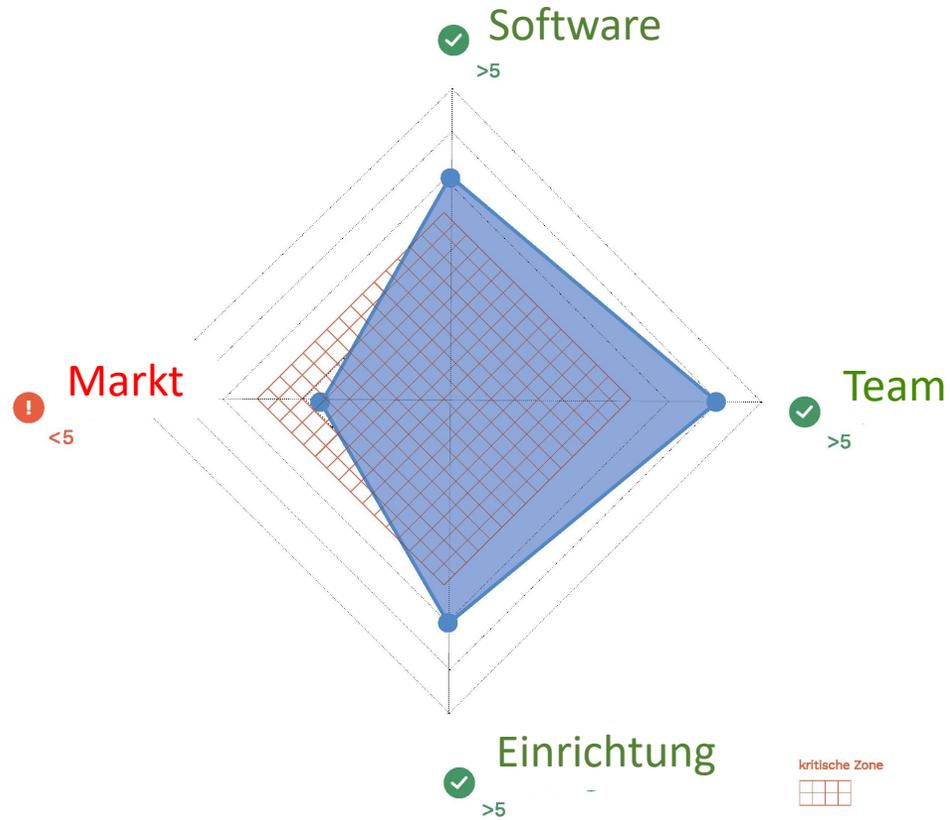
## Die Ersteinschätzung



Handlungsempfehlungen

# Bewertung des Verwertungspotentials

## Das Bewertungstool (Fragenkatalog)

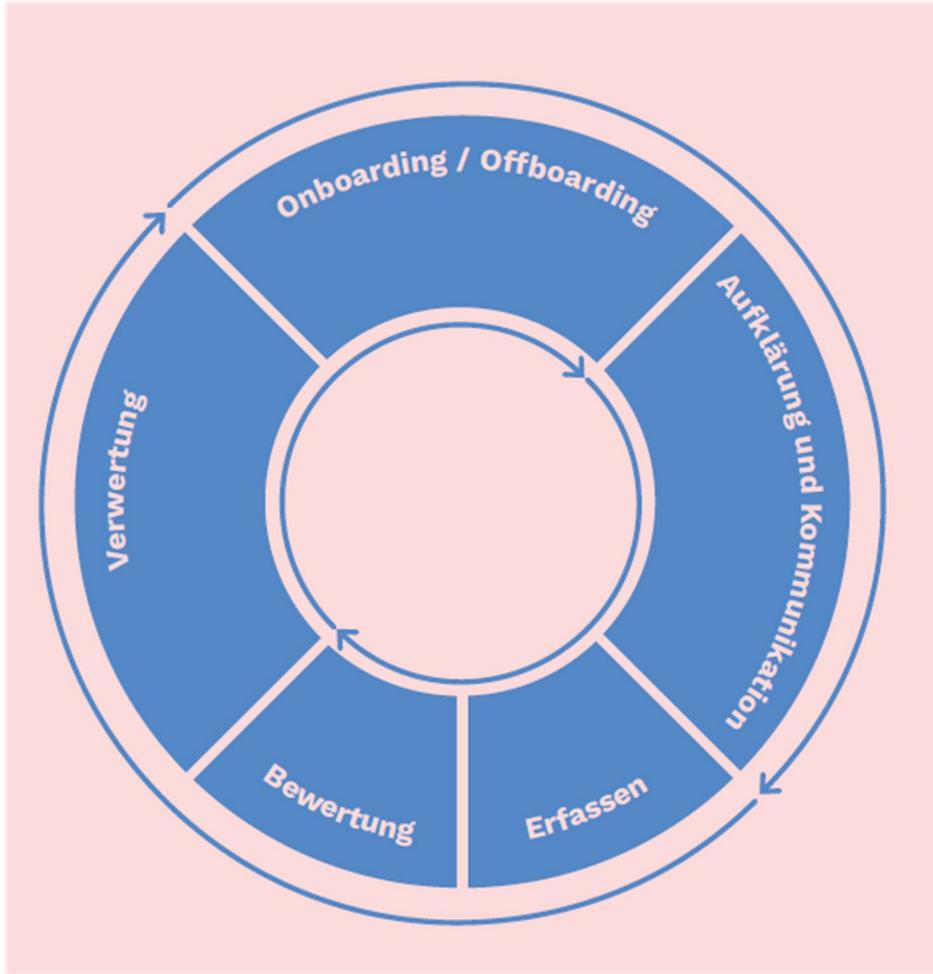


Handlungsempfehlungen

# Agenda

TOP	Thema	Referent
1	Begrüßung und Einleitung	Zahra Saleh
2	Der Methodenbaukasten – die Inhalte	Zahra Saleh
3	Fokusthema 1: Rechtliche Fallstricke	Janine Fischer
<b>15 min Pause</b>		
4	Fokusthema 2: Innovative Transferwege	Lisa Wenzel Michael Jung
5	<b>Good Practice Beispiele aus den Zentren der Projektbeteiligten</b>	<b>Lisa Wenzel</b> <b>Janine Fischer</b>
6	Software Dealdatenbank	Zahra Saleh
7	Take Home Messages	Lisa Wenzel
8	Diskussion und Fragen	alle
<b>Ende des Meetings</b>		

# Good Practices: Strukturen & Prozesse am GFZ



- **Festes Team:** Software Legal Team
- **Etablierter Prozess:** Software Distribution Prozess → digital und (teil-)automatisiert:



# Good Practices: Strukturen & Prozesse am GFZ



<https://makeameme.org/meme/policies-policies-everywhere>



# Good Practices: Strukturen & Prozesse am GFZ

- Helmholtz Software Research Directory
- Helmholtz Software Award
- RSE Meet-Up's
- RSE Ambassadors
- Recommended Practices für Nachhaltige Softwareentwicklung
- Muster für Lizenzdateien, CLA, etc.
- Transferscreenings, Innovationsprojekte (auch für Software)
- Transfersprechstunde
- Videoproduktion

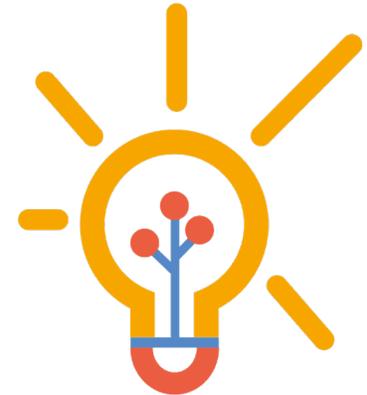
The screenshot displays the Research Software Directory interface. At the top, there's a navigation bar with 'Research Software Directory', a search bar, and tabs for 'Software', 'Projects', and 'Organisations'. The main content area shows the 'GFZ' organization profile, including its logo and contact information. Below this, there are tabs for 'Software (26)', 'Projects (3)', 'Releases (153)', 'Research units (1)', and 'About'. A search filter sidebar on the left allows users to refine results by 'Order by', 'Keywords', 'Program languages', and 'Licenses'. The search results are displayed in a grid, showing three software entries: 'EnPT' (Environmental Mapping and Analysis Program), 'IGMAS+' (3-D forward and inverse modeling software), and 'AROSICS' (automated image co-registration software). Each entry includes a brief description, tags for categories like 'Earth & Environment', and statistics for downloads and citations.

# Good Practices: Strukturen & Prozesse am GFZ

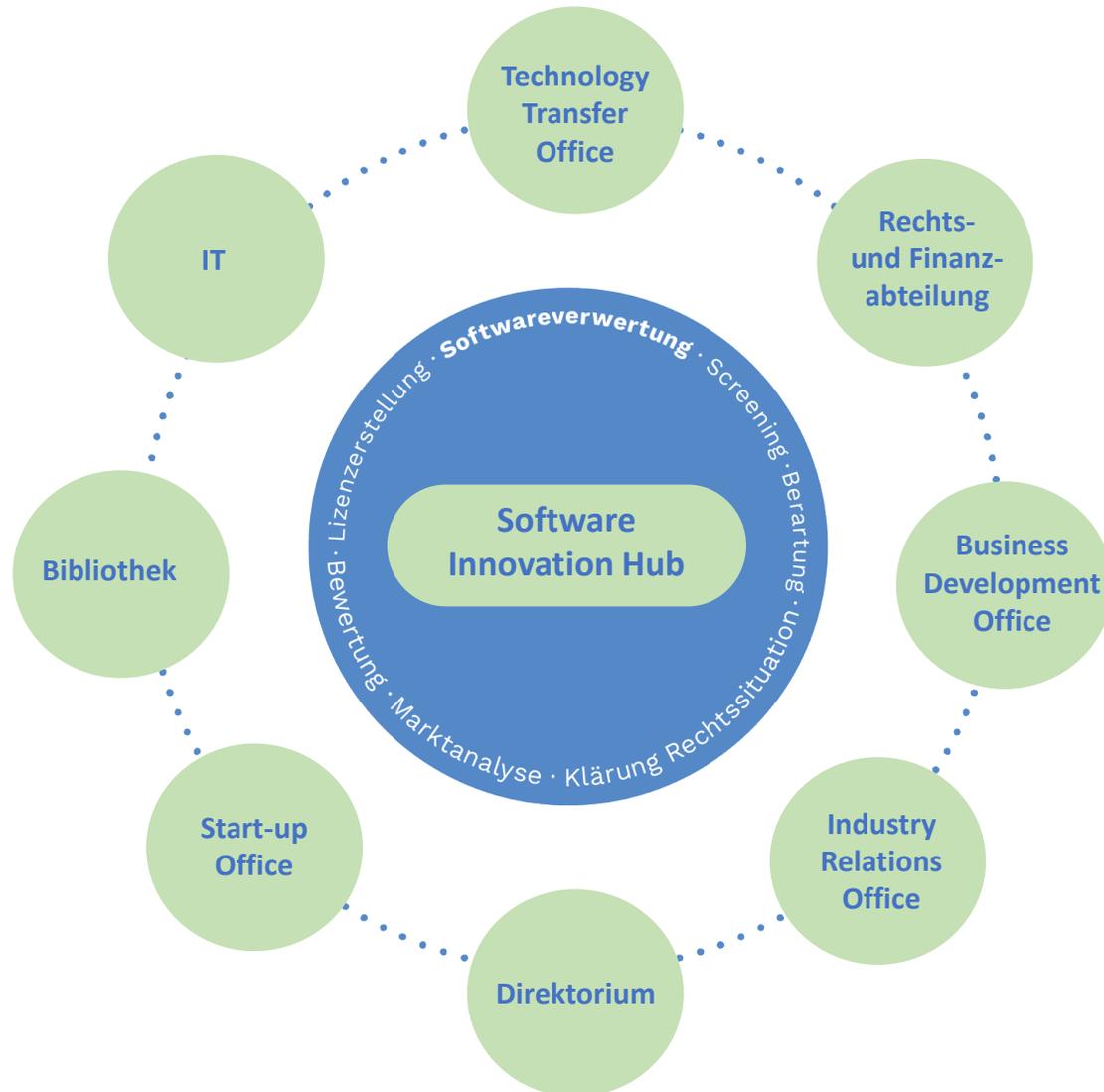
---

## Lessons Learned durch SoftWert:

- Sensibilisierung → Prozesse & Strukturen; Interessenkonflikte Open Science/Transfer
- Forschungssoftware ist eine wertvolle Ressource → Anreizsysteme & Innovative Geschäftsmodelle
- Strategische und systematische Inwertsetzung von Forschungssoftware benötigt professionelle Strukturen, Prozesse und Ressourcen



# Good Practices: Strukturen & Prozesse bei DESY



## Software Innovation Hub

- 1 Person als zentraler Ansprechpartner für Forschungssoftware mit Transfer- und Verwertungspotential (intern und extern)
- Eingebunden in das DESY Innovations- und Technologietransfer Ökosystem

# Software Innovation Hub bei DESY

## INFORMATION



Erfassung von Software



Sensibilisierung

## SOFTWARE VERWERTUNG



Bewertung des Verwertungspotentials



Entscheidungshilfe für Transferwege



Geschäfts-entwicklung

## BERATUNG



Beratung zu Lizenzen



Beratung Software Ausgründungsvorhaben

# Software Innovation Hub bei DESY

## INFORMATION



Erfassung von  
Software



Sensibilisierung

Software Meldeformular

Workshops

- Umgang mit Software IP
- Lizenzierung
- Softwareentwicklung

Software Screening

Informationsplattform

# Software Innovation Hub bei DESY



- Software-Check
- Entwicklerteam
- Marktanalyse
- Industriepartner finden
- Kooperationspartner finden
- Beratung zu Funding Möglichkeiten

- Klärung der Rechtesituation
- Unterstützung bei der Lizenzwahl

- Lizenzerstellung
- Software Verwertung und/oder Veröffentlichung
- Unterstützung beim Aufbau von Services und Dienstleistungen

# Software Innovation Hub bei DESY



Beratung zu  
Lizenzen



Beratung Softwareentwicklung  
Ausgründungsvorhaben

- Mögliche Lizenzsysteme und **Geschäftsmodelle**

- **Open Source** Lizenzen und **FAIR** Anforderungen
- Qualitätsanforderungen der Industrie

Informationsplattform zu

- Lizenzen
- Prozesse, Strukturen und Ansprechpartnern
- Richtlinien und **Handlungsempfehlungen**
- Schulungen und **Workshops**

- Sensibilisierung für die Notwendigkeit **kommerzieller Lizenzen** für kommerzielle Software und Softwareentwicklungsumgebungen

- **Rechtlichen Vorgaben** an Softwareerstellung und Kommerzialisierung

- Akademischen Welt versus kommerzielle Welt
  - **Qualitätsanforderungen**

# Agenda

TOP	Thema	Referent
1	Begrüßung und Einleitung	Zahra Saleh
2	Der Methodenbaukasten – die Inhalte	Zahra Saleh
3	Fokusthema 1: Rechtliche Fallstricke	Janine Fischer
<b>15 min Pause</b>		
4	Fokusthema 2: Innovative Transferwege	Lisa Wenzel Michael Jung
5	Good Practice Beispiele aus den Zentren der Projektbeteiligten	Lisa Wenzel Janine Fischer
6	<b>Software Dealdatenbank</b>	<b>Zahra Saleh</b>
7	Take Home Messages	Lisa Wenzel
8	Diskussion und Fragen	alle
<b>Ende des Meetings</b>		

# Software Dealdatenbank

## SoftWert-Community

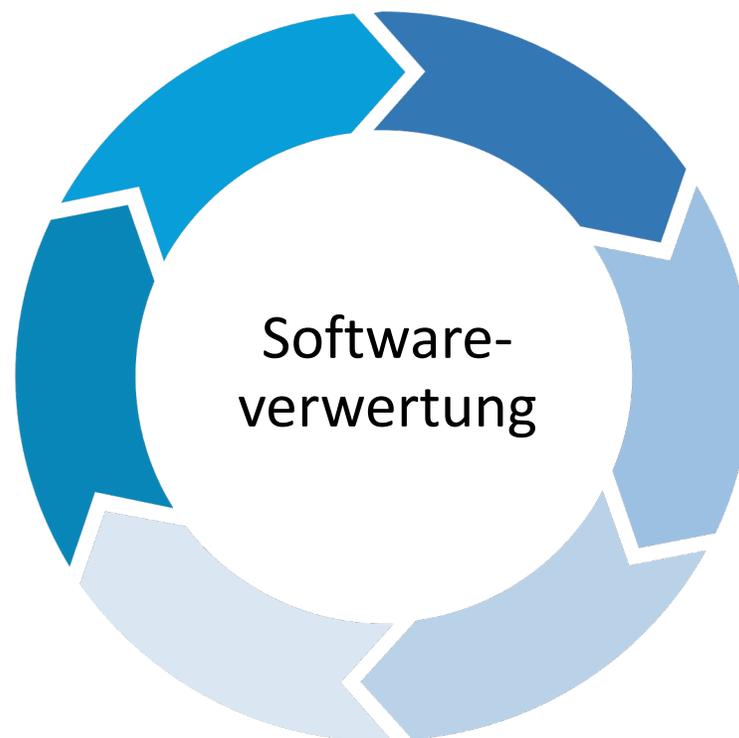
Wissensaustausch zur Softwareverwertung

## “Datenbank”

Charts, Filterfunktion und vieles mehr

## Auswertung und Analyse

Lessons Learned und Besonderheiten



## Online-Umfrage

Jährliche Online-Umfrage von Software-Lizenz-Deals durch DESY ITT

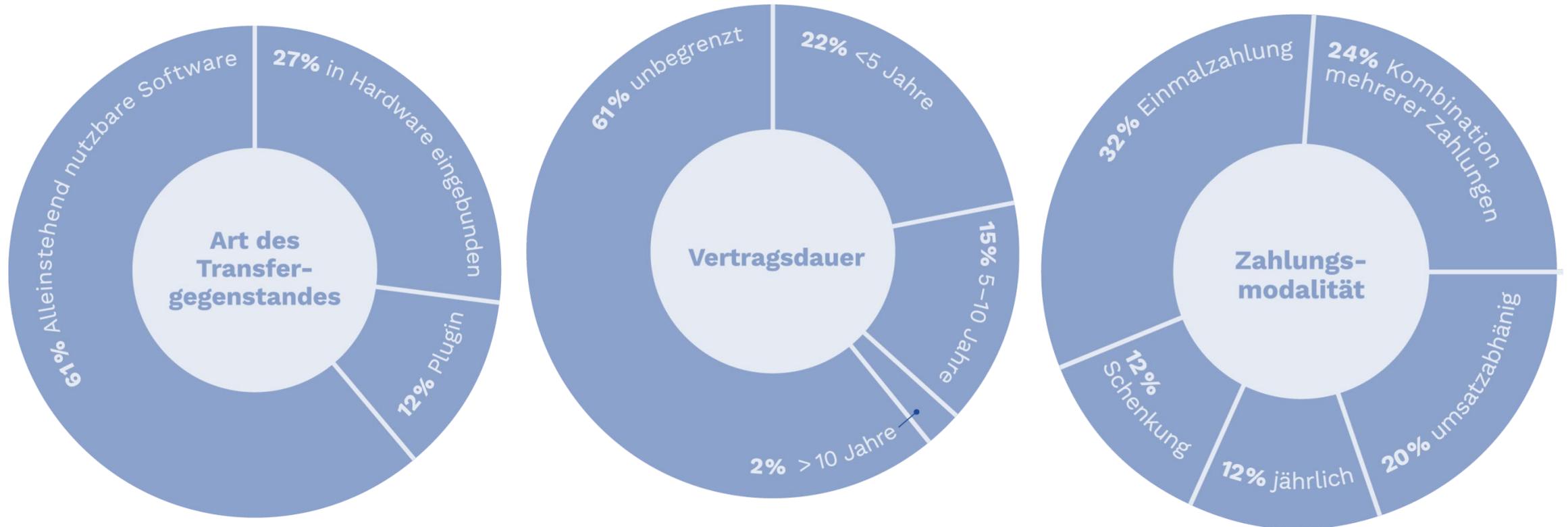
## Zentrale Fragestellungen

Erfassung relevanter Fragestellungen von rechtlichen Grundlagen bis hin zu Einzelheiten der Lizenzierung

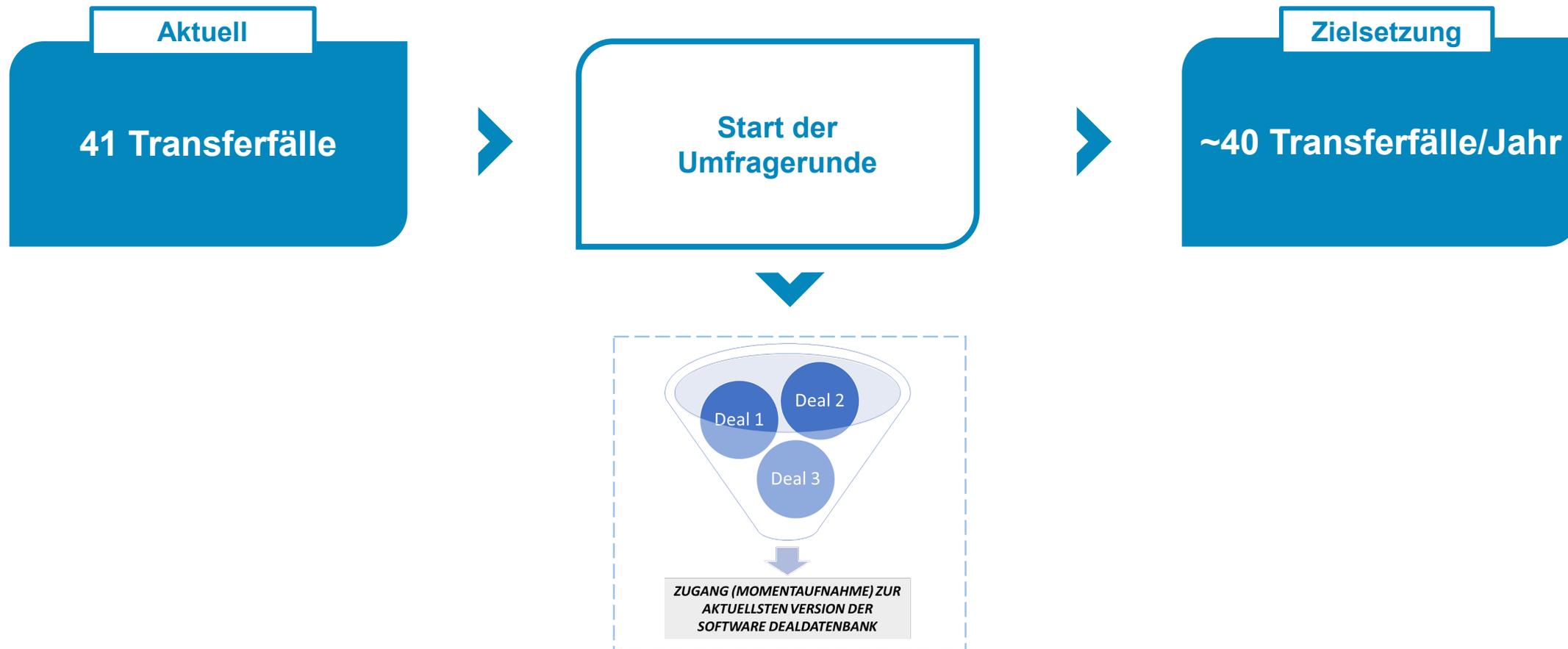
## Anonymisierung & Vernetzung

Anonymisierung der Fälle  
Option der freiwilligen Vernetzung

# Ergebnisse aus der Dealdatenbank



# Aktuell 16 Einrichtungen und ~40 Transferfälle



# Werden Sie Teil der Dealdatenbank und profitieren Sie von den Good Practices!



Link zur Dealdatenbank

oder kontaktieren Sie

DESY



Dr. Janine  
Fischer

E-Mail: [janine.fischer@desy.de](mailto:janine.fischer@desy.de)

# Agenda

TOP	Thema	Referent
1	Begrüßung und Einleitung	Zahra Saleh
2	Der Methodenbaukasten – die Inhalte	Zahra Saleh
3	Fokusthema 1: Rechtliche Fallstricke	Janine Fischer
<b>15 min Pause</b>		
4	Fokusthema 2: Innovative Transferwege	Lisa Wenzel Michael Jung
5	Good Practice Beispiele aus den Zentren der Projektbeteiligten	Lisa Wenzel Janine Fischer
6	Software Dealdatenbank	Zahra Saleh
7	<b>Take Home Messages</b>	<b>Lisa Wenzel</b>
8	Diskussion und Fragen	alle
<b>Ende des Meetings</b>		

# Take Home Messages

---

1. Alle Informationen über die Methoden und Good Practices finden sich auf **softwert.org**
2. Software-Verwertung bedarf einem breiten Spektrum an Methoden → mit richtigen Tools ist das handhabbar
3. Keine Weitergabe von Forschungssoftware ohne Lizenz! → selbst wenn man als TT-Manager keine/wenig Ahnung von Software-Transfer haben sollte
4. Es gibt viele rechtliche Fragestellungen mit unterschiedlicher Komplexität → hole dir einen Profi (FTE)
5. Sei kreativ in der Geschäftsmodellentwicklung, Softwareverwertung bietet viele Möglichkeiten für Impact und Einnahmen → Open Science und Transfer zusammendenken
6. Versteigerung von Strukturen ist wichtig bzw. notwendig

**VIELEN DANK FÜR IHRE AUFMERKSAMKEIT!**