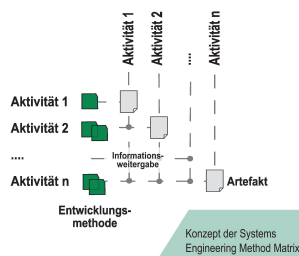


## Systems Engineering Method Matrix: Ein Tool zur anwenderspezifischen Prozess- und Methodengestaltung

Ammersdörfer, T.; Meyer, J.; Inkermann, D.

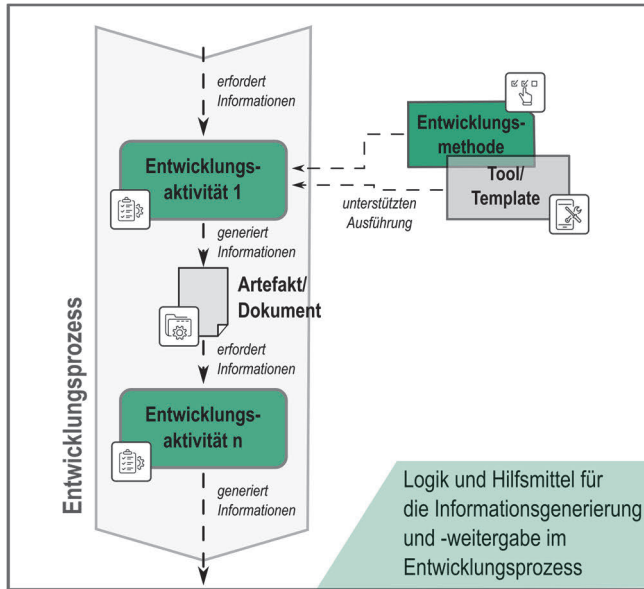


Die Systems Engineering Methoden Matrix (SEMM) unterstützt Entwickler:innen in der Planung von Entwicklungsaktivitäten, bietet Hilfestellung bei der Informationsgenerierung und -weitergabe im Produktentwicklungsprozess und ermöglicht eine Artefakt-orientierte Auswahl kontextbezogener Entwicklungsmethoden. Durch die Aufbereitung und Bereitstellung strukturierter Methodensteckbriefe können die Entwickler:innen neues Methodenwissen erlernen, und Entwicklungsaktivitäten zielgerichtet ausführen.

The Systems Engineering Methods Matrix (SEMM) supports engineers in planning engineering design activities, provides assistance in generating and passing on information in the engineering design process, and enables an artifact-oriented selection of context-related engineering design methods. By preparing and providing structured method profiles, engineers can learn new method knowledge and carry out engineering design activities in a targeted manner.

### Einführung und Problemstellung

Systems Engineering (SE) umfasst verschiedene Entwicklungsaktivitäten wie zum Beispiel das Erfassen von Anforderungen, das Entwickeln von Systemarchitekturen oder die Bewertung von Entwicklungsrisiken [1, 2]. Diese Entwicklungsaktivitäten sind in Prozessen organisiert und aus ihnen gehen verschiedene Informationen zur Beschreibung von Produkten hervor. Abbildung 1 zeigt dazu eine Übersicht, welche die Logik und die Hilfsmittel für die Informationsgenerierung und -weitergabe im Entwicklungsprozess aufzeigt. Für Entwicklungsprojekte werden in Unternehmen häufig weitgehend standardisierte Prozesse genutzt, ohne dass diese vollständig dokumentiert sind. Entwicklungsergebnisse und Informationen sind meist in verschiedenen Dokumenten verteilt und damit teilweise schwer nachvollziehbar. In Abbildung 2 werden exemplarisch unterschiedliche Dokumente wie zum Beispiel die Anforderungsliste oder die Fehlermöglichkeits- und -einflussanalyse dargestellt, die als Informationsspeicher im Entwicklungsprozess dienen können.



**Abbildung 1:** Übersicht zu Logik und Hilfsmitteln für die Informationsgenerierung und -weitergabe im Entwicklungsprozess

ID	Name	Beschreibung	Status	Quelle
1	Systemanforderung	Das System soll den Kunden ein...	geplant	Marketing + Online-Umfrage
2	Produktanforderung	Das System soll die Möglichkeit bieten...	geplant	Marketing + Online-Umfrage
3	Prozessanforderung	Das System soll ein System sein...	geplant	Marketing + Online-Umfrage
4	Benutzeranforderung	Das System soll ein System sein...	geplant	Marketing + Online-Umfrage
5	Leistungsanforderung	Das System soll ein System sein...	geplant	Marketing + Online-Umfrage
6	Benutzeranforderung	Das System soll ein System sein...	geplant	Marketing + Online-Umfrage
7	Leistungsanforderung	Das System soll ein System sein...	geplant	Marketing + Online-Umfrage
8	Benutzeranforderung	Das System soll ein System sein...	geplant	Marketing + Online-Umfrage
9	Leistungsanforderung	Das System soll ein System sein...	geplant	Marketing + Online-Umfrage
10	Benutzeranforderung	Das System soll ein System sein...	geplant	Marketing + Online-Umfrage

System/Produkt/Prozess	Art	Folgen	Ursachen	Kontrollmaßnahmen	empfohlene Abstellmaßnahmen	Verantwortlichkeit	Kontrollmaßnahmen	Verbessertes Zustand
Paketlieferung	Lieferung verspätet	Kundenzufriedenheit	Probleme mit der Strecke ab	Kontrollmaßnahmen: Adressen, Auftragsnummer, RFPZ	empfohlene Abstellmaßnahmen	Verantwortlichkeit	Kontrollmaßnahmen	Verbessertes Zustand
Produkt aus Lager holen	Paket beschädigt	Schlechte Online-Bewertungen	schlechtes Verpackungsmaterial	Kontrollmaßnahmen: Adressen, Auftragsnummer, RFPZ	empfohlene Abstellmaßnahmen	Verantwortlichkeit	Kontrollmaßnahmen	Verbessertes Zustand
Produkt verpacken	Paket beschädigt	Schlechte Online-Bewertungen	schlechtes Verpackungsmaterial	Kontrollmaßnahmen: Adressen, Auftragsnummer, RFPZ	empfohlene Abstellmaßnahmen	Verantwortlichkeit	Kontrollmaßnahmen	Verbessertes Zustand
Paket einladen	Paket beschädigt	Schlechte Online-Bewertungen	schlechtes Verpackungsmaterial	Kontrollmaßnahmen: Adressen, Auftragsnummer, RFPZ	empfohlene Abstellmaßnahmen	Verantwortlichkeit	Kontrollmaßnahmen	Verbessertes Zustand
Paket transportieren	Paket beschädigt	Schlechte Online-Bewertungen	schlechtes Verpackungsmaterial	Kontrollmaßnahmen: Adressen, Auftragsnummer, RFPZ	empfohlene Abstellmaßnahmen	Verantwortlichkeit	Kontrollmaßnahmen	Verbessertes Zustand

**Abbildung 2:** Exemplarische Dokumente als Informationsspeicher im Entwicklungsprozess

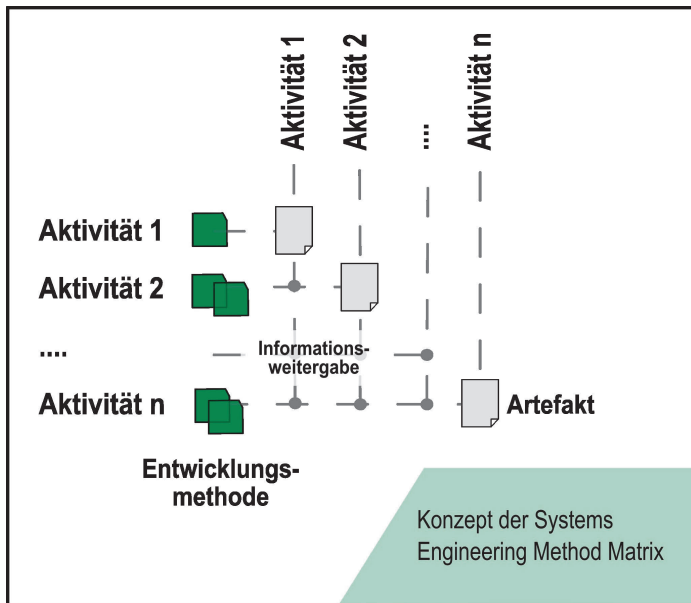
Mit der Veränderung von Produkten besteht die Notwendigkeit, dass Unternehmen ihre Prozesse, aber auch ihre Entwicklungsmethoden anpassen und teilweise erneuern. Daher zielt dieser Betrag darauf ab Entwickler:innen im Umgang mit Veränderungen und den damit einhergehenden Prozess- sowie Entwicklungsmethodenanpassungen zu unterstützen, um den Entwicklungsprozess effektiver und effizienter zu gestalten. Dafür wird in diesem Beitrag, der im Rahmen des Verbundprojektes RePASE (Reflexive Prozessentwicklung und -adaption im Advanced Systems Engineering) entwickelte Systems Engineering Method Matrix (SRMM) - Tool Prototyp vorgestellt.

### Zielsetzung und Forschungsschwerpunkt

Um die Informationsgenerierung und -weitergabe im Entwicklungsprozess zu fördern, sind neben prozessualen Anpassungen auch Methodenanpassungen im Entwicklungsprozess notwendig. Entwickler:innen stehen unter anderem vor der Herausforderung, neue Entwicklungsmethoden zu erlernen oder bestehende Entwicklungsmethoden an einen veränderten Entwicklungskontext anzupassen. Dafür ist prozessorientiertes Methodenverständnis notwendig. Nach /3/ ist eine Methode ein planmäßiges, regelbasiertes Vorgehen nach dessen Vorgabe bestimmte Tätigkeiten auszuführen sind, um ein gewisses Ziel zu erreichen. Daher zielt der Beitrag zum einen darauf ab, Entwickler:innen beim Erlernen von neuem Methodenwissen zu unterstützen. Zum anderen soll mit der Bereitstellung einheitlicher Methodenbeschreibungen die Verständlichkeit der Entwicklungsmethoden verbessert und eine schnellere Anpassung und Vermittlung von Entwicklungsmethoden gefördert werden. Dazu wird die SEMM als Hilfsmittel vorgeschlagen, um Entwicklungsprojekte zu planen, Methodenwissen bereitzustellen, Artefakt-orientiert Entwicklungsmethoden auszuwählen und um Entwicklungsdokumente besser zu managen. Dabei werden insbesondere die Entwicklungsaktivitäten fokussiert und deren Wechselwirkungen untereinander verdeutlicht. Die Aktivitäten des Systems Engineering beziehen sich in diesem ersten Entwurf auf die Systemdefinition und -entwurf des Gesamtsystems.

### Konzept der Systems Engineering Method Matrix (SEMM)

Der Grundaufbau der SEMM orientiert sich an der Matrix-orientierten Methodenbereitstellung nach /4/ und ist ein Hilfsmittel, um Entwicklungsprojekte zu planen, Methodenwissen bereitzustellen und Entwicklungsdokumente zu organisieren. Das Aktivitäts- und Artefakt-bezogene Vorgehen wird in Abbildung 3 schematisch dargestellt.



**Abbildung 3:** Konzept und Aufbau der SEMM

Bei der Planung erforderlicher Entwicklungsaktivitäten mit Hilfe der SEMM, können projektspezifisch die erforderlichen Entwicklungsartefakte (d.h. konkrete Dokumente) definiert und gleichzeitig verschiedene Entwicklungsmethoden zur Unterstützung der Entwicklungsaktivitäten vorgeschlagen werden. Das ist insbesondere für neue Entwickler:innen hilfreich oder bei der Einführung neuer Entwicklungsmethoden relevant.

### Bereitstellung von Methodenwissen und Aufbau der Methodensteckbriefe

Um die oben beschriebenen, benötigten Artefakte erstellen zu können, ist es notwendig, zielgerichtet und effizient die entsprechenden Entwicklungsmethoden anzuwenden, um die notwendigen Ergebnisse zu generieren. Damit Entwickler:innen bereits bekannte Methoden rekapitulieren oder das breite und teils divergente Wissen aus der Literatur zielgerichteter anwenden können, wurden Methodensteckbriefe zur Bereitstellung des Methodenwissens in Anlehnung an /5/ entwickelt. Diese Methodensteckbriefe geben einen kurzen, aber ausreichend präzisen Überblick über die einzelnen zu bearbeitenden Entwicklungsmethoden und gehen unter anderem auf den Zweck der Methode, die Entwicklungssituation, das Vorgehen, das Ergebnis, aber auch auf die Werkzeuge zur Anwendung der Methode ein. Exemplarisch ist ein Methodensteckbrief in der Abbildung 4 und der Abbildung 5 dargestellt. Hierbei handelt es sich um die Methode „Use Case Analyse“, welche auch als „Nutzerorientierte Funktionsmodellierung“ bekannt ist.

<i>Methodenname</i>		
<b>Use Case Analyse (Nutzerorientierte Funktionsmodellierung)</b>		
<i>Schlagworte</i>		
Systemdefinition, Funktionen/ Funktionsstruktur entwickeln, Systemgrenze, Stakeholder, Use-Cases		
<i>Zweck/ Ziel (Kernidee)</i>		<i>Entwicklungssituation (Beabsichtigte Nutzung)</i>
<ul style="list-style-type: none"> <li>▪ Bei der Nutzerorientierte Funktionsmodellierung werden Anwendungsfälle ermittelt, in denen das Produkt genutzt wird. Aus den verschiedenen Anwendungsfällen werden erforderliche Funktionen für das zu entwickelnde Produkt und Anforderungen an das Produkt abgeleitet. (Definition aus SYSTEMIC)</li> <li>▪ Durch das Festlegen von Entwicklungsschwerpunkten und das Abbilden möglicher Anwendungsfälle eines Systems werden das Systemverständnis gefördert und der Umgang mit komplexen Systemen unterstützt. (Zweck aus Lindemann, 2009)</li> </ul>		<ul style="list-style-type: none"> <li>▪ Die nutzerorientierte Funktionsmodellierung findet bezüglich Vorfixierungen auf bereits bekannte Lösungen Anwendung.</li> <li>▪ Ebenfalls bietet sich diese Methode an, um bei technisch komplexen Problemstellungen einen Überblick über nutzerorientierte Anwendungsfälle zu ermöglichen.</li> </ul>
<i>Inputs (erforderlich)</i>	<i>Vorgehen</i>	<i>Outputs (erreichbar)</i>
	<i>Arbeitsschritte der Methode</i>	
<ul style="list-style-type: none"> <li>▪ Systemgrenze</li> <li>▪ Stakeholder</li> </ul>	<ol style="list-style-type: none"> <li>1. Ermitteln der Systemgrenze, welche das System vom Umfeld trennt sowie Ermittlung von Stakeholdern, welche im Zusammenhang mit dem System stehen</li> <li>2. Skizzieren dieser in einem Entwurf</li> <li>3. Sammeln denkbarer Anwendungsfälle bezogen auf die jeweiligen Stakeholder</li> <li>4. Zuordnen der Anwendungsfälle auf den jeweiligen Stakeholder</li> <li>5. Spezifizieren der Anwendungsfälle in Sub-Anwendungsfälle</li> <li>6. Zuordnen der Sub-Anwendungsfällen zu den Anwendungsfällen</li> <li>7. Skizzieren des Use-Case-Modells</li> </ol>	<ul style="list-style-type: none"> <li>▪ Use-Cases</li> </ul>
	<i>Ergebnis der Methode</i>	






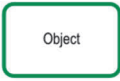


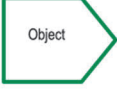
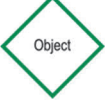







**Abbildung 4: Methodensteckbrief Use Case Analyse (Teil 1)**

<p><b>Formale Aufgabenbeschreibung</b></p>		
<p>Die Use Case Analyse umfasst folgende Aufgaben und Eingangs- sowie Ausgangsinformationen:</p>		
<p><b>Methodenkette (Auswahl von vor- und nachgelagerte Methoden)</b></p>		<p><b>Werkzeuge/ Tools</b></p>
<ul style="list-style-type: none"> <li>▪ Stakeholderanalyse (vorgelagert)</li> <li>▪ Systemkontextanalyse (vorgelagert)</li> <li>▪ Einflussmatrix (nachgelagert)</li> <li>▪ Funktionsdekomposition (nachgelagert)</li> </ul>		<p><b>Einfache Tools:</b> Whiteboard, Tafel, PowerPoint (Vorlage)</p> <p><b>MBSE-Werkzeuge:</b> Enterprise Architect (SysML)</p>
<p><b>Beispiel</b></p>		<p><b>Literatur</b></p>
<p>Die folgende Abbildung zeigt eine Use Case Analyse für einen Staubsaugerroboter:</p>		<p>Lindemann, 2009</p> <p>SYSTEMIC Karten</p>

**Abbildung 5: Methodensteckbrief Use Case Analyse (Teil 2)**

Alle in der SEMM verwendeten Methodensteckbriefe sind strukturell gleich aufgebaut, vergleiche Abbildung 4 und Abbildung 5 für die folgende Erklärung. Jeder Methodensteckbrief beginnt mit einem Namensfeld indem der Titel der Methode festgelegt wird. Das zweite Informationsfeld umfasst Schlagworte, um diese Methode auch hinsichtlich anderer Methoden, Aktivitäten oder Artefakten durchgängig verorten zu können. Ebenfalls ist es möglich Methoden auf diese Weise zu suchen. Ein simples Beispiel ist, wenn der Nutzer eine Systemdefinition durchführen möchte und die Fragestellung erscheint, welche Schritte dafür notwendig sind. Aus Abbildung 4 lässt sich erkennen, dass auch die Use Case Analyse ausgegeben werden würde. Das nächste Feld beinhaltet den Zweck bzw. das Ziel des Methodensteckbriefs. Hier kann der Nutzer prägnant in ein bis zwei Stichpunkten erkennen, ob diese Methode für ihn hilfreich ist. Eine weitere wichtige Frage für einen jeden Nutzer ist, ob dieser Methodensteckbrief zur aktuellen Entwicklungssituation passt. Diese Entwicklungssituation wird ebenfalls auf das Wesentliche reduziert und im nächsten Informationsfeld festgehalten. Anschließend folgt eine neue Tabellenzeile in der im linken Bereich der Abbildung 4 benötigte Inputs, ggf. von vorherigen durchgeführten Methoden festgehalten werden, jene von Nöten sind um die aktuelle Methode des Methodensteckbriefs durchzuführen. Im rechten Bereich werden die Outputs aufgelistet, welche mit dieser Methode erzeugt werden. Zwischen Inputs und Outputs ist das Vorgehen der Methode in Form von aufeinanderfolgenden nummerierten Arbeitsschritten und als exemplarisches verallgemeinertes Ergebnis festgehalten. Ebenfalls wurde für die Grundstruktur der Methodensteckbriefe eine Formale Aufgabenbeschreibung eingeführt. Diese ist in Abbildung 5 zu erkennen. Die Formale Aufgabenbeschreibung kann als eigenständig entwickelte grafische Sprache zur Verdeutlichung und Visualisierung der Aufgaben einer Entwicklungsmethode verstanden werden. Diese ist ebenfalls eine Alternative zum Lesen der oben beschriebenen Arbeitsabfolge zur Ergebniserarbeitung des Methodensteckbriefs.

Die Notation der Formalen Aufgabenbeschreibung hingegen ist in Abbildung 6 zu sichten. Es gibt fünf wesentliche Hauptoperatoren mit denen sich sämtliche Abfolgen des Methodensteckbriefs beschreiben lassen. Diese werden durch ein markantes grafisches Symbol, vergleiche Abbildung 6, beschreiben. Die Symbole sind zum einen der Sketch-Operator, welcher das Erstellen einer Skizze, Abbildung o.ä. erfordert. Ein weiteres Symbol ist der List-Operator, welcher das Auflisten bzw. das Erfassen von Daten erfordert. Weiterhin existiert der Assign-Operator, welche die gesammelten Daten zuordnet. Der Connect-Operator verbindet einzelne Operatoren. Letztlich bleibt der Gate-Operator, welche eine Schleife einleitet. Ebenfalls gibt es zwei weitere benötigte grafische Operatoren, vergleiche unterer Bereich der Abbildung 6. Diese sind zum einen der Aufgabenrahmen der Methode, welcher abstrakt gedacht die Systemgrenze beschreibt. Zum anderen der Verbindungspfeil, welcher die einzelnen Operationen vernetzt und unerlässlich für den Input sowie den Output über die Systemgrenzen ist.

	Sketch	List	Assign	Connect	Gate
Symbol					
Beschreibung	Skizzen-Operation, welche das Erstellen einer Darstellung erfordert.	Listen-Operation, welche das auflisten von Daten initiiert.	Zuordnen-Operation, welche das zuordnen von diversen Daten beschreibt.	Verknüpfen-Operation, welche das verbinden von Operationen umschreibt.	Tor-Operation, welche den Beginn von Schleifen umschreibt.
Object					
Value/Property					
Weiteres	Aufgabenrahmen der Methode 	Beschreibung → 			

**Abbildung 6:** Notation der Formalen Aufgabenbeschreibung der SEMM

Um den aufgeworfenen Methodensteckbrief in einer Ergebniskette zu verorten bzw. integrieren zu können, existiert ein weiteres Informationsfeld, das der Methodenkette. Hier sind vor- und nachgelagerte Methoden zu sichten. So kann effizient ein Anschluss an die Methode oder einzelne vorgelagerte Methoden identifiziert werden, um einen benötigten Input zu generieren oder den Output weiter zu verarbeiten. Ein weiteres Feld umfasst Werkzeuge bzw. Tools. Hier werden die Bereiche einfache Tools und Spezialwerkzeuge unterschieden. Die einfachen Tools sind meist beim Anwender frei oder günstig verfügbar und liefern bei korrekter Abarbeitung der Methode ein solides verwertbares Ergebnis. Die Spezialwerkzeuge sind eine spezifische, meist monetär anspruchsvolle Alternative, bei der eine stärkere Vernetzung und Koexistenz der Ergebnisse möglich ist. Abschließend wird ein grafisches Beispiel der Ergebnisse des Methodensteckbriefs dargestellt. Dieses Beispiel ist zum Querverständnis aller Methodensteckbriefe durchgängig dasselbe und behandelt die Entwicklung eines Staubsaugroboters. Zur Untermauerung der erarbeiteten Inhalte eines jeden Steckbriefs wird im letzten Informationsfeld eine Auswahl an verwendeter Literatur dargestellt. Diese kann bei tiefergehenden Informationsbedarf ebenfalls helfen.



## Funktionsübersicht der SEMM

Für eine langfristig zielgerichtete Umsetzung der methodischen Mechanismen der SEMM in eine digitale Variante, sind einige wesentliche Funktionen von Nöten. Diese lassen sich übersichtlich in die Bereiche technischer Zugriff auf die SEMM, Erstellung und Verwaltung unterschiedlicher Projekte, Erstellung und Änderung der SEMM, Erstellen und Pflegen von Entwicklungsaktivitäten, Erstellen und Pflegen von Methodenbeschreibungen, Erstellen und Pflegen von Entwicklungsartefakten und Unterstützung der Nutzung, unterteilen. Folglich sind die einzelnen Funktionen der Kategorien aufgelistet:

### Zugriff auf SEMM:

- Das Frontend der SEMM soll über übliche, aktuelle Browser aufgerufen werden. Vor der Nutzung müssen Nutzernamen und Passwörter überprüft werden. Für den SEMM-Prototyp muss keine Nutzerverwaltung umgesetzt werden.
- Die Anzeige und Nutzung des Frontends muss mit PC und mobilen Endgeräten ohne Einschränkungen der Darstellung und Editierung möglich sein.

### Erstellung und Verwaltung unterschiedlicher Projekte:

- Es müssen unterschiedliche SEMM als Projekte verwaltet werden können. Über die GUI müssen unterschiedliche Projekte angelegt und ausgewählt werden können.
- Für ein Projekt muss ein Titel, Ersteller:in und ein Erstellungsdatum angegeben werden.
- Änderungen in einem Projekt müssen mit Zeitangabe und Namen gespeichert werden. Die Änderungshistorie muss einsehbar sein.
- In einem Projekt müssen die bestehenden Methodensteckbriefe und Vorlagen verwendet und verknüpft werden können. Die Verwendung muss in den jeweiligen Steckbriefen und Vorlagen gespeichert werden („wird verwendet in Projekt XY“).

### Erstellung und Änderung einer SEMM:

- Für die Erstellung einer neuen SEMM sollen im Anzeigefenster erste Aktivitätsblöcke (links) mit Blindtexten einschließlich Artefakten und Methoden angezeigt werden.
- Für die Vervollständigung und Anpassung einer SEMM müssen bestehende Entwicklungsaktivitäten editiert (Titel, Beschreibung) werden können. Wenn eine neue Entwicklungsaktivität erstellt wird, müssen im Anzeigefenster neue Zeilen und Spalten sowie Blöcke für Methoden und Artefakte ergänzt werden.

- Wenn eine Entwicklungsaktivität geändert wird, muss die/ der Nutzer:in auf die Überprüfung der Input- und Output-Artefakte und Methoden hingewiesen werden.

#### Erstellen und Pflegen von Entwicklungsaktivitäten:

- Für eine Entwicklungsaktivität müssen immer ein Titel, eine Beschreibung sowie Input- und Output-Artefakte und Methoden angegeben werden.
- Die Erstellung und Pflege muss über eine GUI via Texteingabe und Auswahllisten möglich sein.
- Für jede Entwicklungsaktivität muss mindestens ein Input- und ein Output-Artefakt angegeben werden. Bei Nichterfüllung ist ein Hinweis zu erzeugen.
- Für jede Entwicklungsaktivität muss mindestens eine Methode angegeben werden. Bei Nichterfüllung ist ein Hinweis zu erzeugen.
- Input- und Output-Artefakte sowie Methoden müssen aus einer Liste ausgewählt werden können. Bei der Eingabe soll im Idealfall automatisiert ein Vorschlag zur Vervollständigung erzeugt werden (Wünschanforderung). Wenn neue Artefakte erzeugt werden, sind diese in die bestehende Liste aufzunehmen.
- Bestehende Entwicklungsaktivitäten sollen geändert werden können. Änderungen sollen anhand einer Änderungshistorie (Datum, Nutzer:in) nachvollziehbar sein.

#### Erstellen und Pflegen von Methodenbeschreibungen:

- Methoden müssen durch Texte und Abbildungen beschrieben werden können. Die Eingabe der Beschreibungen muss über eine GUI via Freitext und Upload von Abbildungen möglich sein.
- Die Methodenbeschreibung umfasst folgende Attribute des oben beschriebenen Methodensteckbriefs, siehe Abbildung 4 und Abbildung 5.
- Alle Methoden müssen in einer Liste (Datenbank) archiviert werden.
- Es muss möglich sein, weitere Attribute für die Methodenbeschreibung zu ergänzen.
- Methodenbeschreibungen sollen als .pdf-Dokument heruntergeladen werden können (Wunschforderung).
- Bestehende Methodenbeschreibungen sollen geändert werden können. Änderungen sollen anhand einer Änderungshistorie (Datum, Nutzer:in) nachvollziehbar sein.

#### Erstellen und Pflegen von Entwicklungsartefakten:

- Für jedes Entwicklungsartefakt muss ein Titel und eine Kurzbeschreibung angegeben werden. Optional können Dokumente (.pdf, .docx, .doc, .xlsx, .xls) hinterlegt werden. Die Eingabe der

Beschreibungen muss über eine GUI via Freitext und Upload von Dokumenten möglich sein.

- Alle erzeugten Entwicklungsartefakte müssen in einer Liste (Datenbank) gespeichert werden.

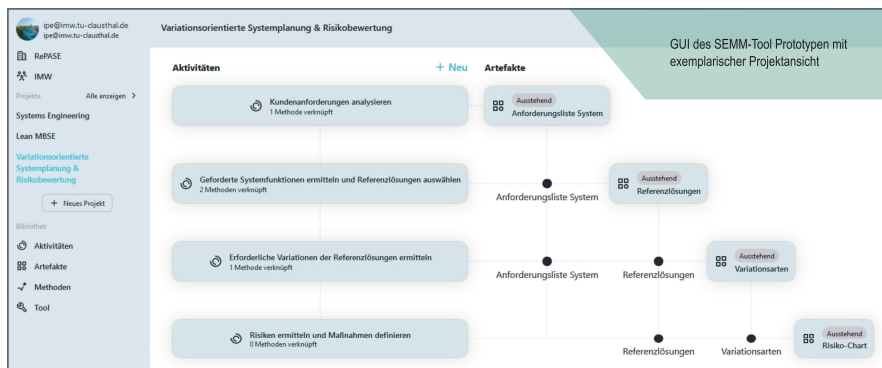
Unterstützung der Nutzung:

- Im Hauptfenster sollen z.B. via Mouse-Over ergänzende Informationen (z.B. Kurzbeschreibungen für Entwicklungsaktivitäten, Methoden und Entwicklungsartefakt) angezeigt werden können.
- Bei Auswahl von Entwicklungsaktivitäten, Methoden oder Entwicklungsartefakte sollen im unteren Fenster detaillierte Informationen angezeigt werden.
- Die Erstellung und Änderung von Entwicklungsaktivitäten, Methoden oder Entwicklungsartefakten soll jeweils über ein Fenster ermöglicht werden. Das Fenster wird nach Auswahl des jeweiligen Objektes geöffnet.

Mit diesen Hauptfunktionen wird die digitale webbasierte Umsetzung der SEMM im nächsten Abschnitt teilweise vorgestellt.

## Vorstellung des SEMM-Tool Prototypen

Die im SEMM-Tool hinterlegte Funktion ermöglicht es, Abhängigkeiten zwischen Entwicklungsaktivitäten und Dokumenten festzulegen (vgl. Abbildung 7).



**Abbildung 7:** GUI des SEMM-Tool Prototypen mit exemplarischer Projektsicht

Dazu wird definiert, welche Information aus einer vorherigen Aktivität oder einem Dokument wiederverwendet werden sollen, wie beispielsweise die Wiederverwendung von Systemanforderungen oder Funktionen für die Konzeptentwicklung oder auch die Fehleranalyse. Wir verzichten dabei bewusst auf formale Beschreibungen, wie sie im Model-based Systems Engineering

vorgeschlagen werden. Damit ist keine Einführung neuer oder komplexer Entwicklungswerkzeuge erforderlich. Die zugrundeliegende Logik der Abhängigkeiten zwischen Entwicklungsaktivitäten, Artefakte und Entwicklungsmethoden, ermöglicht es, trotzdem eine hohe Durchgängigkeit und Rückverfolgbarkeit von Informationen innerhalb eines Entwicklungsprojektes zu erreichen (vgl. Abbildung 8). Das prototypische Tool steht als browserbasierte Lösung zur Verfügung und kann auf die Anforderungen unterschiedlicher Unternehmen angepasst werden. Über die Zeit entwickelt sich mit der praktischen Anwendung innerhalb dieses Tools eine unternehmensspezifische Wissensbasis, die verschiedene Entwicklungsaktivitäten, Templates für die Erstellung von Dokumenten aber auch Methodenwissen umfasst. Das Tool konnte im Rahmen des Verbundprojektes RePASE als prototypische Lösung umgesetzt und mit Praxispartnern erprobt werden. Zudem wurden mehr als 40 Methodensteckbriefe erarbeitet, die im SEMM-Tool zur Verfügung stehen.

Kundenanforderungen analysieren

1. Allgemein 2. Input-Artefakte 3. Output-Artefakte 4. Methoden (optional)

**Zugewiesene Artefakte**

Name

Anforderungsliste System

**Artefakte zuweisen** + Neues Artefakt

Name

Artefakt 01

Dokumentation Kundenanfrage

Anforderungsliste System

Morphologischer Kasten und Referenzlösungen

Referenzlösungen

Artefakt #dcf6

Variationsarten

Risiko-Chart

Verknüpfung von Entwicklungsaktivitäten und Artefakten bei der Projektplanung

→ Weiter ✓ Speichern und weiter

**Abbildung 8:** Verknüpfung von Entwicklungsaktivitäten und Artefakten bei der Projektplanung am Beispiel „Kundenanforderungen analysieren“

## Zusammenfassung und Ausblick

Die SEMM wurde entwickelt, um Entwickler:innen in der Durchführung komplexer Entwicklungsprozessaktivitäten konsistent sowie flexibel zu unterstützen und dadurch die Effizienz im Entwicklungsprozess zu steigern. Die SEMM beinhaltet verschiedene Aktivitätsebenen, welche (Ergebnis-)Artefakte als notwendige Inputs und Outputs erzeugen. Diese werden wiederum durch die Methodenanwendung erzeugt. Um eine einheitliche und präzise Erzeugung von Methodenergebnissen bzw. Artefakten zu gewährleisten, wurden vereinheitlichte Methodensteckbriefe erstellt. Damit die SEMM schnell und zielgerichtet digital umgesetzt werden kann, wurden notwendige technische und logische Funktionen zusammengetragen. Diese wurden in einem digitalen Prototyp umgesetzt und sind für die Bearbeitung von verschiedenen Anwenderprojekten bereit. Letztlich ist der Prototyp der SEMM durch ein wachsendes Portfolio von Entwicklungsmethoden kontinuierlich erweiterbar. Durch das Einpflegen weiterer Projekte, können weitere Entwicklungsbedarfe aufgezeigt werden. Derzeit wurde sich im SEMM Prototypen auf den Bereich der Systemdefinition und -entwurf des Gesamtsystems fokussiert. Zukünftig sollen auch die weiteren Bereiche im Produktentwicklungsprozess wie beispielsweise die Systemintegration oder die Systemverifikation aufbereitet und entsprechende Methodensteckbriefe zur Verfügung gestellt werden. Damit die SEMM kontinuierlich weiterentwickelt werden kann, ist die Erprobung und Anwendung mit weiteren unterschiedlichen Anwendungspartnern hilfreich.

## Danksagung

Die vorgestellten Forschungsergebnisse sind Teil des Projekts RePASE - Reflexive Prozessentwicklung und -adaption im Advanced Systems Engineering. Dieses Projekt wird vom Bundesministerium für Bildung und Forschung (BMBF) im Rahmen des Programms "Innovationen für die Produktion, Dienstleistung und Arbeit von morgen" (02J19B149) gefördert und vom Projektträger Karlsruhe (PTKA) geleitet. Die Verantwortung für den Inhalt dieser Veröffentlichung liegt bei den Autoren.

## Literatur

- /1/ ISO15288 (2015): ISO15288:2015 Systems and software engineering. System life cycle processes.
- /2/ INCOSE (2015), INCOSE-TP-2003-002-04: Systems Engineering Handbook: A guide for system life cycle processes and activities, International Council on Systems Engineering (INCOSE), San Diego, USA.
- /3/ Birkhofer, H.; Kloberdanz, H.; Berger, B.; Sauer, T.: Cleaning up design methods-describing methods completely and standardised, In DS 30: Proceedings of DESIGN 2002, the 7th International Design Conference, Dubrovnik, 2002, <https://api.semanticscholar.org/CorpusID:54516885>
- /4/ Wach, J. J., Problemspezifische Hilfsmittel für die Integrierte Produktentwicklung, München: Hanser 1994 (Konstruktionstechnik München, Band 12). Zugl. Dissertation 1993, Technische Universität München
- /5/ Lindemann, U.: Methodische Entwicklung technischer Produkte: Methoden flexibel und situationsgerecht anwenden. VDI-Buch. Berlin Heidelberg: Springer-Verlag, 3., korrigierte Auflage, 2009, ISBN 978-3-642-01422-2