

Model-based Systems Engineering in der Schnittstelle zwischen Entwicklung und Produktion

Rollenprofile zur erfolgreichen Umsetzung des MBSE

F. Rusch, H. Hage, F. Mantwill

ZUSAMMENFASSUNG Dieser Ansatz zur Rollenzuteilung im Model-based Systems Engineering (MBSE) unterscheidet zwischen prozessorientierten und modellorientierten Rollen, um Effizienz und Interoperabilität im MBSE zu steigern. Durch die Integration bestehender Kompetenzrahmen und Best Practices wird eine gezielte Nutzung digitaler Werkzeuge unterstützt. Herausforderungen bestehen in der organisatorischen Umsetzung und Qualifikation. Zukünftige Forschung sollte die praktische Anwendung evaluieren.

STICHWÖRTER

Advanced Systems Engineering, Informationsmanagement, Arbeitsorganisation

Role profiles for the successful implementation of MBSE

ABSTRACT This approach for role allocation in Model-based Systems Engineering (MBSE) differentiates between process-oriented and model-oriented roles to enhance efficiency and interoperability in MBSE. By integrating existing competency frameworks and best practices, a targeted use of digital tools is enabled. Challenges arise in organizational implementation and qualification. Future research should evaluate practical application.

1 Einleitung

Erhöhte Qualitätsanforderungen sowie technischer Fortschritt führen in der Produktentwicklung zu zunehmend komplexeren Systemlösungen, wie cyberphysischen Systemen. Die Absicherung sowie Nachverfolgbarkeit dieser Systeme stehen im Vordergrund. Das Model-based Systems Engineering (MBSE) bietet hier einen modellbasierten Ansatz, der das Herunterbrechen von Komplexität ermöglicht. Das MBSE findet in der Praxis aber nur selten Anwendung oder wird nur als Dokumentationsmethode genutzt, weshalb das MBSE als interoperable Entwicklungsmethode noch nicht vollständig ausgeschöpft wird [1]. Ein Grund hierfür sind zum Beispiel zu generische und zu umfassende Kompetenzfelder für Systems Engineers, die eine komplexitätsgerechte Umsetzung des MBSE durch ausführende Akteure erschweren [2]. Entwicklungsunternehmen müssen Rollenprofile für Spezialisten oft ohne methodische Unterstützung selbstständig definieren. Ein zielführendes und strukturiertes Kompetenzmanagement ist aber nötig, um komplexe Aufgaben erfolgreich zu lösen [3].

Ziel dieses Beitrags ist die Konzeption eines Rollenmodells, das prozess- und modellorientierte Aufgaben im MBSE strukturiert abbildet. Im Mittelpunkt steht ein Referenzrahmen, der Entwicklungsunternehmen bei der Einordnung und Integration MBSE-relevanter Rollen in bestehende Strukturen unterstützt. Das Modell adressiert insbesondere die zunehmende Systemkomplexität sowie Anforderungen an klare Zuständigkeiten, modellbasierte Zusammenarbeit und toolgestützte Interoperabilität.

2 Literaturanalyse

2.1 Methodische Vorgehensweisen im Systems Engineering

Die Literaturanalyse erfolgte auf Basis einer gezielten Auswahl relevanter Fachliteratur, die zunächst auf einschlägigen, in der Forschung etablierten Arbeiten beruhte und anschließend durch die Analyse der Referenzlisten sowie durch ergänzende, begriffsbezogene Recherchen in Google Scholar erweitert wurde.

Systems Engineering (SE) ist ein wissenschaftlich fundierter Ansatz, der durch eine systematische und interdisziplinäre Vorgehensweise die Entwicklung und den Betrieb komplexer Systeme ermöglicht. Dabei bilden in der technischen Entwicklung des SE insgesamt 14 technische Prozesse die Grundlage, welche nach Incose [4] definiert sind, **Tabelle 1**.

Die Strukturierung und Zusammenhänge der technischen Prozesse lassen sich grob über das V-Modell erklären, welches zum Beispiel nach VDI/VDE 2206 [5] definiert ist. Im V-Modell wird der grundsätzliche Zusammenhang zwischen den technischen Prozessen vereinfacht aufgezeigt. Dabei wird der Entwicklungsprozess von kontinuierlichen Verifizierungs- und Validationszyklen abgesichert [4]. Ein umfassenderer Zusammenhang zwischen den technischen Prozessen des SE ist im Kapitel 2.3.5 des Incose SE Handbuchs [4] erläutert. SE wird vor allem bei der Entwicklung komplexer technischer Systeme eingesetzt, da diese besonders von Komplexitätsreduktion, Absicherung und Nachverfolgbarkeit profitieren.

Neben dem SE nach Incose [4] gibt es weitere Vorgehensweisen im SE. Ein Ansatz ist das Generic Systems Engineering

Tabelle 1. Technische Prozesse im Systems Engineering, abgeleitet von INCOSE SE Handbook [4].

Technischer Prozess im SE	Beschreibung
Geschäfts- und Missionsanalyse	Analysiert das Geschäftsumfeld oder die Missionsziele, um die Notwendigkeit eines neuen Systems zu identifizieren.
Stakeholderbedarfe und Anforderungen	Erfasst und analysiert die Bedürfnisse und Erwartungen der Stakeholder.
Systemanforderung	Übersetzt die Stakeholder-Anforderungen in konkrete technische Systemanforderungen.
Architekturgestaltung	Entwickelt eine strukturierte Systemarchitektur, welche die Anforderungen erfüllt.
Entwurf	Erarbeitet eine detaillierte technische Spezifikation auf Basis der Architektur und legt physikalische Eigenschaften der Systemkomponenten fest.
Systemanalyse	Bewertet verschiedene Systemlösungen mithilfe von Modellen, Simulationen und quantitativen Analysen.
Umsetzung	Implementiert die Systemelemente konform zur Spezifikation.
Integration	Führt die Systemkomponenten schrittweise zusammen und testet ihre Interaktion und Funktionalität.
Verifizierung	Prüft durch Tests, Analysen, Inspektionen und Simulationen, ob das entwickelte System die spezifizierten Anforderungen erfüllt.
Übergabe	Übergibt das entwickelte System an den Endnutzer oder Betreiber. Umfasst Schulungen, Installation und die Sicherstellung der Betriebsbereitschaft.
Validierung	Stellt sicher, dass das System die Bedürfnisse der Stakeholder erfüllt und für den vorgesehenen Zweck geeignet ist.
Betrieb	Unterstützt den laufenden Betrieb des Systems und sorgt für eine zuverlässige Nutzung.
Wartung	Stellt die langfristige Funktionsfähigkeit des Systems sicher. Beinhaltet Wartung, Reparaturen, Updates und Modernisierungen.
Entsorgung	Behandelt die Außerbetriebnahme und Entsorgung des Systems unter Berücksichtigung von Umwelt- und Sicherheitsaspekten sowie Recycling in Wiederverwendung von Komponenten.

(GSE), das Methoden für nachhaltige Systemarchitekturen und agiles Systemdesign verbindet sowie Werkzeuge zur Modellierung und Analyse komplexer technischer Zusammenhänge berücksichtigt [6]. Ein weiterer relevanter Ansatz ist das Advanced Systems Engineering (ASE), das den Fokus auf die Interaktion zwischen Systementwicklungsprozessen und deren Kontexten legt und Methoden der Prozessmodellierung sowie Bewertungsansätze integriert, um eine dynamische Anpassung an veränderte Anforderungen zu ermöglichen. Dabei werden ganzheitliche Aspekte, wie Mensch, Technik und Organisation berücksichtigt [7]. Außerdem liefern Beiträge im Lösungsraum zunehmende Konkretisierungen, wie etwa zum kontinuierlichen und dynamischen Ableiten von Nachhaltigkeitsanforderungen [8] und der gezielten Transformationen von Unternehmen zur Förderung der Integrierbarkeit des ASE [9].

Ebenfalls häufiger eingesetzt wird das agile SE, das iterative und inkrementelle Entwicklungsansätze in das traditionelle SE integriert. Es kombiniert agile Prinzipien mit der Stringenz von SE-Methoden, um schnell auf Veränderungen in den Anforderungen reagieren zu können, die Entwicklungszeiten zu verkürzen und die Qualität zu verbessern [10]. Auch gibt es etliche weitere SE-Ansätze, die sich auf einzelne Aspekte fokussieren. Zum Beispiel das Sustainable SE, welches sich auf die Berücksichtigung

nachhaltiger Aspekte auf Basis der Sustainable Development Goals (SDGs) oder partielle Fokussierungen und Formalisierungen für das SE fokussiert, wie *Rusch et al.* [11] mit der Formalisierung eines nachhaltigen Systemkontexts und dem Erfassen von Nachhaltigkeitseinflüssen.

2.2 Model-based Systems Engineering

MBSE erweitert klassische SE-Ansätze durch die Verwendung von Modellen als zentrale Informationsquelle und kann als modellbasierter Ansatz des SE gesehen werden [4]. Als Modellierungssprache wird oft die Systems Modeling Language (SysML) verwendet. Es gibt verschiedene Frameworks und Methoden, die eine strukturierte Vorgehensweisen für spezifische Anwendungsfälle des MBSE bieten.

Einer der bekanntesten ist der RFLP-Ansatz von Dassault. RFLP bietet eine Integration von „Requirements, Functional, Logical and Physical“-Modellen, was eine durchgängige Modellierung sowie Nachverfolgbarkeit über alle Entwicklungsphasen hinweg erlaubt [12]. IBM „Harmony SE“ verfolgt einen iterativen Ansatz, bei dem Anforderungen, Design und Verifikation eng miteinander gekoppelt werden, um eine durchgängige Systementwicklung zu gewährleisten [13]. Ein weiterer Ansatz ist IBM

Tabelle 2. Methodische Vorgehensweisen im MBSE und Modellarchitektur.

Name Vorgehensweise	Plattform / Tool	Gliederung Modellarchitektur	Quelle(n)
IBM Harmony SE	IBM Rhapsody	Requirements Analysis Functional Analysis Design Synthesis	[13]
IBM RUP SE	IBM Rational Developer	Inception Phase Elaboration Phase Construction Phase Transition Phase	[14]
Dassault RFLP	Cameo Systems Modeler	Requirements Analysis Functional Analysis Logical Analysis Physical Analysis	[12]
ARCADIA	Capella	Customer Operational Need Analysis System Need Analysis Logical Architecture Physical Architecture Building Strategy	[15]
CONSENS	iQUAVIS	Environment Application scenarios Requirements Functions Active Structure Shape Behavior	[16]
INCOSE OOSEM	All SysML Tools	Analyse Needs Define System Requirements Define Logical Architecture Synthesize Allocated Architectures Optimise and Evaluate Alternatives Validate and Verify System	[17]
Vitech STRATA	CORE	Requirements Analysis Functional Behavior Analysis Architectural Synthesis Validation & Verification	[18]
WSAF	CORE	Strategic Domain Operational Domain System Domain Analysis Domain	[19]
JPL State Analysis	n. a.	Requirements Analysis Functional Analysis Components Analysis	[20]

„RUP SE“, eine Erweiterung des Rational Unified Process, die sich auf das Management komplexer Anforderungen und eine inkrementelle Entwicklung konzentriert [14]. „Arcadia“ legt den Fokus auf Architekturmodellierung und differenziert klar zwischen funktionalen und operativen Sichten, um die Komplexität der Systementwicklung zu reduzieren [15]. „Consens“ integriert funktionale und physikalische Aspekte in einem konsistenten Rahmen für die mechatronische Systementwicklung [16].

Mit IncoSE „Oosem“ wird ein objektorientierter Ansatz verfolgt, der modellbasiertes Denken mit objektorientierten Paradigmen verknüpft [17]. Vitech „Strata“ bietet Ansätze für die Modellierung und Verwaltung von Anforderungen, Verhaltensmodellen und Architekturen [18]. Für eine spezifische Anwendung in der militärischen Systementwicklung eignet sich das Weapon Systems Architecture Framework (WSAF) [19]. Die „JPL State Analysis“ der NASA legt den Schwerpunkt auf die modellbasierte Zustandsanalyse und die Ableitung effektiver Steuerungsstrategien [20]. Es bestehen noch weitere bekannte Ansätze, die im Rahmen dieser Analyse aufgrund fehlender Zugriffsberechtigungen oder geringfügig eingestufteter Relevanz nicht berücksichtigt wurden.

Die genannten MBSE-Vorgehensweisen wurden in den ausgewiesenen Quellen in **Tabelle 2** auf die Gliederung einer Modellarchitektur hin analysiert. Die Ergebnisse dieser Analyse sind in **Tabelle 2** wiedergegeben.

2.3 Kompetenzen und Rollenverteilung im Systems Engineering

Da das SE komplexe Fragestellungen aus unterschiedlichen Disziplinen vereint, werden in den jeweiligen Prozessbereichen spezifische Kompetenzen besonders benötigt. Hierfür definiert das IncoSE Systems Engineering Competency Framework (SECF) [21] verschiedene Fähigkeiten für Systems Engineers. Im SECF Assessment Guide [22] werden die schwerpunktmäßigen benötigten Kompetenzen aus dem SECF unter anderem mit den technischen Prozessen des SE nach IncoSE gemappt.

Ein weiterer Ansatz für die Fokussierung von Kompetenzen im SE ist die Einführung von Rollen, welchen Aufgabenbereiche zugeordnet sind. *Sarah Sheard* definierte in ihrem Paper „Twelve Systems Engineering Roles“ [23] insgesamt zwölf Rollen für das SE, welche durch eine Aufgabenverteilung und Zuordnung von Schnittstellen die Durchführung der technischen Prozesse unterstützen sollen. In einem weiteren Beitrag von *Sheard* [24] wird der Wert dieser zwölf Rollen diskutiert. Dabei werden folgende vier Kernvorteile gesehen: Die Einflussnahme auf das Produkt vor dem Kunden steigt durch die Rollenzuteilung; es wird strukturierter gearbeitet; Probleme können auf mehreren Ebenen besser definiert werden und die Risiken der Produktentwicklung werden reduziert.

Gräßler et al. greifen den Ansatz der zwölf Rollen 2019 wieder auf [25] und schlagen auf Basis von *Sheard*, angereichert mit industrieller Erfahrung und akademischen Kompetenzen, einen Rahmen von 15 Rollen im SE unter dem Namen „RM-SEA“ vor. Diese sollen helfen, einer Organisation zeitgemäß eine Grundlage zu liefern, um organisationsinterne Strukturen so anzupassen, dass erfolgreich SE betrieben werden kann. In einem weiteren Beitrag von *Gräßler et al.* [26] wurden diese Rollen hinsichtlich ihrer Aufgabenbereiche mit SE Artefakten gemappt. In **Tabelle 3** sind die Rollen von *Sheard* [23] und *Gräßler et al.* [25] gegenübergestellt.

Im MBSE sind nur geringfügige Rollenverteilungen bekannt. So findet eine Rollenzuteilung im IBM-RUP-SE-Ansatz statt, indem zwischen Systems Engineers, Architects, Developers und Testers unterschieden wird [14]. In dem Rollenmodell von *Gräßler et al.* werden dem Modeling Engineer die Aufgaben des MBSE zugeordnet [25]. Ein umfassendes Rollenmodell zur Fokussierung der Aufgabenbereiche des MBSE ist darüber hinaus nicht bekannt.

2.4 Fazit Literaturanalyse

Aus der Literaturanalyse geht hervor, dass die bestehenden Ansätze zur Rollenverteilung im SE zwar wertvolle Rahmenbe-

Tabelle 3. Rollendefinitionen für Systems Engineering, nach *Sheard* [23] und *Gräßler et al.* [25].

Rollenverteilung nach <i>Sheard</i> (1996)	Rollenverteilung nach <i>Gräßler et al.</i> (2019)
1. Requirements Owner	1. Modelling engineer
2. System Designer	2. System architect
3. System Analyst	3. System interface manager
4. Validation/Verification Engineer	4. Technical manager
5. Logistics/Ops Engineer	5. Leader
6. Glue Among Subsystems	6. Stakeholder interface manager
7. Customer Interface	7. Requirements engineer
8. Technical Manager	8. Life Cycle engineer
9. Information Manager	9. Implementation engineer
10. Process Engineer	10. Validation and verification engineer
11. Coordinator	11. Security engineer
12. Classified Ads SE	12. Process owner
	13. Configuration manager
	14. Information manager
	15. Entrepreneur

dingungen für die Aufgaben- und Schnittstellenzuweisung bieten, jedoch zentrale Lücken aufweisen. Während bisherige Ansätze, wie von *Sheard* [23] oder *Gräßler et al.* [25] beschrieben, eine detaillierte Strukturierung der Rollen im SE schaffen, fehlt es an einer klaren Brücke zum MBSE. Vor allem das Zusammenspiel zwischen Prozessen und der Modellierung bleibt rollenseitig unzureichend definiert.

Im MBSE werden spezifische Methoden und Frameworks zwar hervorgehoben, die eine durchgängige Modellierung unterstützen, die Rollendifferenzierung ist aber weit weniger ausgearbeitet. IBM RUP SE nennt zwar grundlegende Rollen, beschränkt sich jedoch auf oberflächliche Definitionen.

Die Analyse zeigt, dass bisherige Rollenmodelle, etwa nach *Sheard* [23] und *Gräßler et al.* [25], und modellbasierte Ansätze im SE bislang unzureichend miteinander verzahnt sind, sodass das Potenzial des MBSE in der praktischen Umsetzung nicht vollständig ausgeschöpft wird. Auch die Arbeit von *Hage* [27] unterstreicht die Komplexität in der praktischen Integration von SE, weshalb unzureichend definierte Rollenmodelle die Anwendung von MBSE zusätzlich erschweren. Ein Rahmen, der sowohl modellbasierte als auch prozessorientierte Perspektiven integriert und rollenspezifische Interaktionen klar definiert, bleibt unerreicht und bietet Raum für methodische Weiterentwicklungen.

3 Ansatz zur Systematisierung einer Rollenzuteilung im Systems Engineering

3.1 Übersicht des Ansatzes

Der folgende Ansatz unterscheidet zwischen prozess- und modellorientierten Rollen. Während prozessorientierte Rollen primär für die Steuerung und Umsetzung der Geschäftsprozesse verantwortlich sind, konzentrieren sich modellorientierte Rollen auf die systematische Modellierung und methodische Unterstüt-

zung zum Erreichen des Ziels dieser Prozesse. Trotz ihrer unterschiedlichen Schwerpunkte ergänzen sich beide Gruppen, um eine ganzheitliche, effiziente und nachhaltige Entwicklung komplexer technischer Systeme zu sichern. Diese Unterscheidung soll ermöglichen, dass Modelle über eine gezielte Kommunikationsschnittstelle entstehen können und durch die Ergebnisse die Prozesse unterstützen. **Bild 1** zeigt die grundsätzliche Strukturierung des Ansatzes.

Durch die fokussierte Expertise dieser Rollen soll ein erhöhter Erkenntnisgewinn modellseitig umgesetzt und in den Prozess übertragen werden. Die Rollen ergänzen somit die bereits bestehenden prozessorientierten Rollen, welche auf die Orchestrierung von Prozessen auf verschiedensten Abstraktionsebenen fokussiert sind und ermöglichen somit einen Expertenaustausch zwischen Prozess und Modell. Der Ansatz verfolgt das Ziel, bislang ungenutzte Potenziale des MBSE zu adressieren. Er bietet einen Rahmen, um etablierte Entwicklungsmethoden mit dem Aufbau und der Pflege von Modellen zu verknüpfen und fachliche Expertise gezielt in die Modellebene zu integrieren. Zudem soll das Rollenmodell die Integration weiterer modellbasierter Methoden, wie CAD, CAM und CAE, systematisch unterstützen. Ziel ist es, die Interoperabilität im MBSE zu verbessern, um modellbasierte Absicherungsketten effizienter zu gestalten – insbesondere bei der Nutzung mehrerer, vernetzter Tools. Die Definition der Rollen und die Fokussierung auf die technischen Prozesse des SE nach [4] werden in den folgenden Kapiteln vorgestellt. Dabei ist es wichtig zu erwähnen, dass die vorgestellten Rollen eine generalisierte Referenz darstellen und Rollenabweichungen im Rahmen von Individuallösungen in Abhängigkeit von Systemarten und Projekt- sowie Unternehmensgröße sinnvoll sein können.

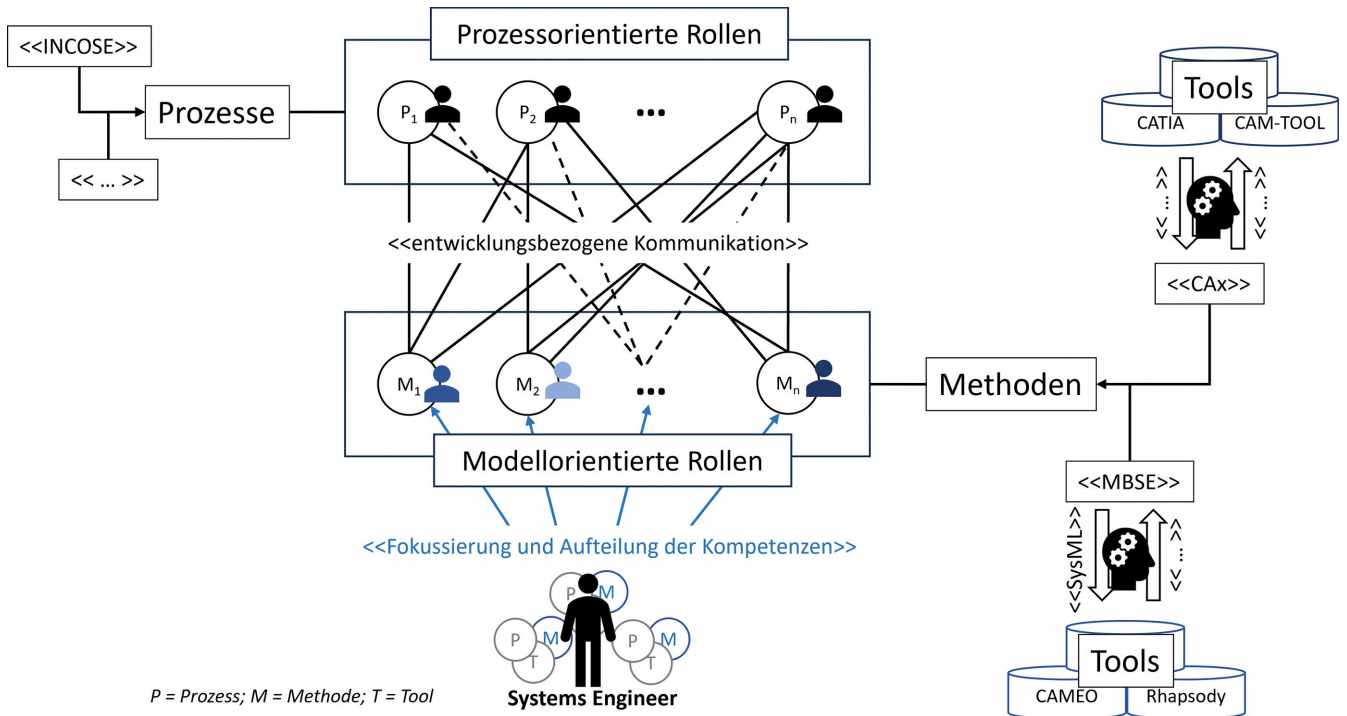


Bild 1. Rollenunterteilung des Systems Engineers in prozessorientierte- und modellorientierte Rollen basierend auf Prozesse, Methoden und Tools. Grafik: HSU Hamburg

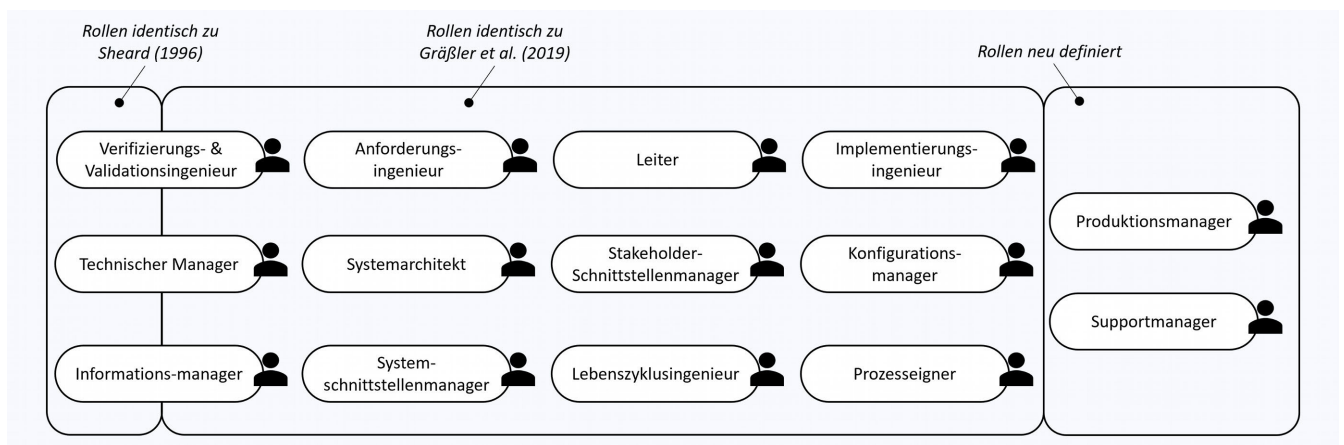


Bild 2. Definition prozessorientierter Rollen für Systems Engineering. Grafik: HSU Hamburg

3.2 Definition prozessorientierter Rollen

Im Folgenden werden prozessorientierte Rollen für die Umsetzung des SE definiert. Diese Rollen sind in **Bild 2** dargestellt.

Neben zwei neu definierten Rollen basieren sämtliche weitere Rollen auf Sheard [23] und Gräßler et al. [25], da diese Rollendefinitionen auf umfangreiche Analysen basieren und als äußerst sinnvoll angesehen werden. Folgend werden kurze Beschreibungen der Rollen vorgenommen.

Der Verifizierungs- & Validationsingenieur stellt sicher, dass das System die Anforderungen erfüllt und die Stakeholder zufriedenstellt. Er plant und führt Tests zum Systemverhalten durch und analysiert identifizierte Abweichungen. Der Technische Manager trifft technologiebezogene Entscheidungen, bewertet und wählt technische Lösungen aus und koordiniert Entwicklungsaktivitäten. Er sorgt für eine effiziente Zusammenarbeit und

identifiziert frühzeitig technische, organisatorische oder personelle Herausforderungen. Der Informationsmanager sammelt, verwaltet und verteilt relevante Daten, um fundierte Entscheidungen zu unterstützen. Er koordiniert disziplinübergreifende Informationen und berücksichtigt dabei die Schnittstellen. [23, 25]

Zudem übernimmt der Informationsmanager in diesem Rollenmodell die Aufgaben des Security Engineers, um modernen Strukturen einer Data Governance zu entsprechen.

Der Anforderungsingenieur definiert, verwaltet und überprüft Systemanforderungen. Er achtet auf Konsistenz zwischen den verschiedenen Anforderungsebenen und stellt sicher, dass das Systemdesign die Anforderungen erfüllt. Der Systemarchitekt entwickelt die Systemarchitektur, wählt Lösungsansätze aus und legt die logische Struktur des Systems fest. Der Systemschnittstellenmanager koordiniert die Schnittstellen zwischen Systemelementen, um ein optimales Gesamtsystem zu gewährleisten, vor allem

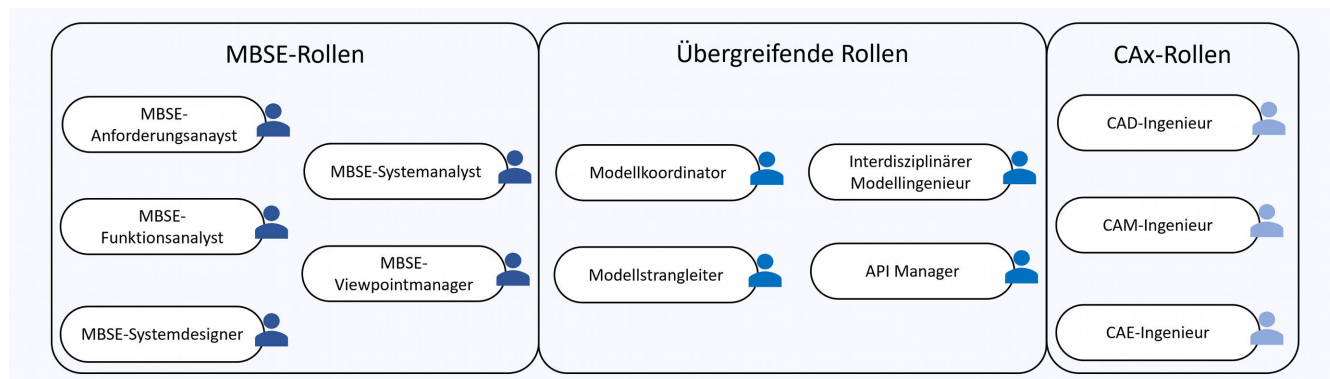


Bild 3. Definition modellorientierter Rollen für Systems Engineering. Grafik: HSU Hamburg

bei steigender Komplexität und der Nutzung externer Komponenten. Der Leiter führt und motiviert das Team, delegiert Aufgaben und sorgt für eine effiziente Zusammenarbeit. Der Stakeholder-Schnittstellenmanager stellt sicher, dass die Bedürfnisse der Stakeholder erfasst, in Anforderungen überführt und kontinuierlich angepasst werden. Er berücksichtigt dabei die Kundenperspektive und kommuniziert Projektentscheidungen. Der Lebenszyklusingenieur integriert eine ganzheitliche Lebenszyklussicht in die Entwicklung und achtet auf Nachhaltigkeitsaspekte, Systemintegrität und langfristige Nutzbarkeit. Der Implementierungsingenieur setzt Systemelemente gemäß Designvorgaben um. Er identifiziert und vermeidet dabei frühzeitig ineffiziente Lösungen. Der Konfigurationsmanager verwaltet alle projektrelevanten Ausgaben (wie Dokumente, Artefakte), stellt ihre Korrektheit sicher und achtet auf die technische, wirtschaftliche und finanzielle Balance. Der Prozesseigner entwickelt, überwacht und optimiert Prozesse im SE. Er passt diese an Projekt- und Unternehmensanforderungen an und stellt ihre Effizienz sicher. [25]

Der Produktionsmanager stellt sicher, dass Systeme effizient und qualitätsgerecht gefertigt werden. Er koordiniert Fertigungsprozesse, optimiert Produktionsabläufe und sorgt für die Einhaltung technischer Vorgaben. Im Unterschied zu klassischen Rollen wie dem Produktionsplaner oder dem Leiter Produktion ist der Produktionsmanager im Kontext des SE deutlich früher in die Produktentwicklung eingebunden. Er bringt produktionsrelevante Randbedingungen gezielt in frühe Entwurfs- und Architekturentscheidungen ein und wirkt dadurch integrativ auf Systemdesign und Herstellbarkeit. Diese Rolle wurde von den Autoren ergänzt, da im SE wiederholt Defizite in der frühzeitigen Berücksichtigung fertigungstechnischer Anforderungen beobachtet wurden. Der Produktionsmanager fungiert daher als Bindeglied zwischen Systementwicklung und Fertigung, vor allem in MBSE-getriebenen Prozessen, in denen modellbasiert bereits in frühen Phasen abgesichert wird, ob technische Konzepte auch herstellbar und wirtschaftlich umsetzbar sind.

Der Supportmanager sorgt bereits in der Entwicklungsphase dafür, dass Anforderungen aus Betrieb, Wartung und Systementsorgung systematisch berücksichtigt werden. Er plant effiziente Wartungskonzepte, minimiert Betriebskosten und stellt sicher, dass ein technisches System auch in der Nutzungsphase betreut und wartbar bleibt. Im Unterschied zu klassischen Service- oder After-Sales-Funktionen ist der Supportmanager im Kontext des SE proaktiv in den frühen Entwicklungsphasen verankert. Durch diese frühe Einbindung kann er individuelle Serviceanforderun-

gen in die Architektur und Systemanforderungen überführen, vor allem bei hochgradig kundenspezifischen Lösungen. Diese Rolle wurde ebenfalls von den Autoren ergänzt, da der steigende Individualisierungsgrad komplexer technischer Systeme die Planung und Absicherung der Supportfähigkeit bereits in der Entwicklungsphase erfordert.

3.3 Definition modellorientierte Rollen

Die in diesem Beitrag dargestellten modellorientierten Rollen (siehe Bild 3) stützen sich auf zentrale Modellierungsaktivitäten etablierter MBSE-Methoden, darunter der RFLP-Ansatz [12] und das IncoSe-Oosem-Vorgehen [17]. Diese Methoden strukturieren den Entwicklungsprozess in funktionale Aufgabenbereiche wie Anforderungsanalyse, Funktionsmodellierung, Architekturdefinition und Systembewertung. Aus diesen Aufgaben lassen sich differenzierte Rollen ableiten, die im Sinne einer arbeitsteiligen Modellverantwortung gestaltet sind. Ergänzend werden in diesem Modell bestehende Rollenprofile aus dem CAx-Umfeld integriert, um deren Bedeutung für modellbasierte Entwicklungsprozesse systematisch zu berücksichtigen. Durch ihre funktionale Einbindung in den SE-Kontext können bestehende Kompetenzen gezielter mit MBSE-Anforderungen verknüpft und technische Schnittstellen effektiver gestaltet werden. Zudem berücksichtigt das Modell auch übergeordnete Rollen, welche domänenübergreifende Koordination, methodische Konsistenz sowie die Toolintegration unterstützen. Die Darstellung in **Bild 3** differenziert die Rollen anhand ihrer primären Aufgabenbereiche innerhalb des MBSE.

Der MBSE-Anforderungsanalyst strukturiert, dokumentiert und verwaltet Anforderungen, um eine konsistente und nachvollziehbare Grundlage für die Systementwicklung zu schaffen. Er modelliert Abhängigkeiten zwischen Anforderungen, stellt deren Vollständigkeit und hierarchische Struktur sicher und analysiert Anwendungsfälle, um funktionale Erwartungen zu präzisieren. Der MBSE-Funktionsanalyst beschreibt und analysiert die funktionale Architektur eines Systems. Er modelliert Abläufe, Interaktionen und Abhängigkeiten zwischen Funktionen, um eine schlüssige und effiziente Struktur sowie die Anforderungserfüllung und Konsistenz zur physischen Struktur zu gewährleisten. Der MBSE-Systemdesigner entwickelt die Systemarchitektur, indem er die Struktur, das Verhalten und die Schnittstellen des Systems definiert. Er beschreibt funktionale Abläufe, Kommunikationsflüsse sowie Zustandsübergänge und sorgt für eine klare Verbindung zwischen logischen und physikalischen Elementen.

Der MBSE-Systemanalyst bewertet und optimiert die Systemarchitektur hinsichtlich Struktur, Funktionalität und Verhalten. Er analysiert Abläufe und Systemreaktionen auf unterschiedliche Eingangsszenarien, modelliert Wechselwirkungen zwischen Systemkomponenten und überprüft Konsistenz sowie Effizienz. Der MBSE-Viewpointmanager koordiniert systematisch den Einsatz unterschiedlicher Modellierungssichten (wie funktionale, logische, physische oder verhaltensbezogene Sichten) innerhalb des MBSE. Er definiert Sichtenmodelle, klärt deren Abgrenzung und Abhängigkeiten und sorgt für die Synchronisierung zwischen diesen Perspektiven. Zudem sichert er die semantische Konsistenz und trägt dazu bei, dass inkonsistente oder widersprüchliche Modellinhalte frühzeitig erkannt und systematisch aufgelöst werden.

Der CAD-Ingenieur erstellt und verwaltet digitale 3D-Modelle von Systemkomponenten und Gesamtstrukturen. Er verantwortet die geometrische Ausgestaltung des Produkts auf Basis funktionaler und architektonischer Vorgaben aus dem MBSE. Er stellt sicher, dass die Konstruktionen die technischen Anforderungen erfüllen. Dabei achtet er auf Integrationsfähigkeit der Komponenten in das Gesamtsystem, fertigungsgerechtes Design und die Einhaltung technischer Schnittstellen. Zudem optimiert er Bauteilgeometrien hinsichtlich Fertigbarkeit, Materialeffizienz und struktureller Integrität. In enger Abstimmung mit dem MBSE-Systemdesigner und dem Modellkoordinator sorgt der CAD-Ingenieur dafür, dass konstruktive Lösungen konsistent in die Systemmodelle rückgeführt und Änderungsbedarfe systematisch adressiert werden.

Der CAM-Ingenieur überführt digitale Konstruktionsmodelle in Fertigungsprogramme für automatisierte Produktionsprozesse. Dabei stellt er sicher, dass relevante Fertigungsanforderungen bereits frühzeitig in den Entwicklungsprozess einfließen, um eine nahtlose Umsetzbarkeit sicherzustellen. Er optimiert Fertigungsstrategien hinsichtlich Wirtschaftlichkeit, Maschinenverfügbarkeit und Qualitätssicherung. In enger Abstimmung mit dem CAD-Ingenieur und dem Modellkoordinator gewährleistet er, dass CAM-relevante Modellinhalte korrekt parametrisiert, toolseitig kompatibel und prozesssicher umgesetzt werden können. Er trägt somit dazu bei, dass modellbasierte Entwicklungsdaten ohne Medienbrüche in den realen Fertigungsprozess überführt werden können und bildet eine Schnittstelle zwischen digitalem Produktmodell und realem Produktionssystem.

Der CAE-Ingenieur führt simulationsgestützte Analysen durch, um das Verhalten von Systemen unter virtuellen Bedingungen zu bewerten. Er modelliert und simuliert physikalische Eigenschaften, wie strukturelle Belastungen, thermische Einflüsse, Strömungsverhalten oder elektromagnetische Effekte, und nutzt diese Analysen zur Unterstützung der Systemvalidierung. Dabei achtet er auf eine konsistente Verknüpfung der Simulationsmodelle mit den zugrunde liegenden Systemarchitekturen sowie auf die Rückführung relevanter Erkenntnisse in das Gesamtsystemmodell. In enger Zusammenarbeit mit dem MBSE-Systemanalysten und dem Modellkoordinator stellt er sicher, dass Modelle abgestimmt, kompatibel und nachvollziehbar integriert werden. Er leistet somit einen entscheidenden Beitrag zur frühzeitigen Absicherung von Anforderungen, vor allem bei Aspekten, die sich nur schwer physisch testen lassen.

Der Modellkoordinator plant, steuert und überwacht modellbasierte Aktivitäten über verschiedene Disziplinen hinweg. Er sorgt für konsistente Modellierungsstrategien, legt methodische Standards fest und koordiniert die Modellqualitätssicherung über

Modellversionen, Toolumgebungen und Entwicklungsphasen hinweg. Zudem stellt er sicher, dass Modellierungsziele regelmäßig mit den Projektzielen abgeglichen und bei Bedarf angepasst werden. Der Modellstrangleiter verantwortet einen zusammenhängenden Modellierungsstrang, der aus mehreren interagierenden Modellen bestehen kann. Er koordiniert dabei das Zusammenspiel der Modellbestandteile und stellt die Zielerreichung sicher.

Der Interdisziplinäre Modellingenieur agiert als vernetzende Schnittstelle zwischen disziplinspezifischen Modellierungsansätzen, vor allem aus MBSE, CAD, CAE und angrenzenden Bereichen. Er besitzt ein breites methodisches Verständnis der jeweiligen Modelltypen und stellt sicher, dass diese konsistent verknüpft und interoperabel nutzbar gemacht werden. Dabei berücksichtigt er domänenspezifische Anforderungen, Datenformate und Qualitätssicherungsrichtlinien und vermittelt zwischen Fachbereichen hinsichtlich Modellverwendung und -integration.

Der API-Manager verantwortet die datentechnische Integration zwischen verschiedenen Modellierungs- und Entwicklungswerkzeugen. Er konzipiert, implementiert und wartet APIs sowie Datenpipelines, die den sicheren, konsistenten und performanten Austausch von Informationen erlauben. Besonderes Augenmerk liegt auf der Verbindung heterogener Toolandschaften und der Etablierung standardisierter Schnittstellenformate (wie XMI, FMI, ReqIF). Gerade im MBSE ist die oft fehlende Interoperabilität zwischen MBSE-spezifischen Modellierungswerkzeugen und angrenzenden CAx-Systemen eine zentrale Herausforderung. Der API-Manager adressiert diese Lücke, indem er fehlende Standards durch projektspezifische Schnittstellenlösungen kompensiert und so die durchgängige Modellvernetzung ermöglicht. Darüber hinaus bewertet er technische Einschränkungen und gestaltet Übergangsstrategien für nicht standardisierte Integrationen.

3.4 Zusammenspiel prozess- und modellorientierter Rollen in den technischen SE-Prozessen

Prozessorientierte Rollen sind direkt in den technischen Prozessen des SE verankert und tragen die Verantwortung für deren Durchführung und Steuerung. Dazu gehören Rollen wie der Anforderungsingenieur, der Systemarchitekt oder der Lebenszyklusingenieur, die sicherstellen, dass Systemanforderungen erhoben, umgesetzt und über den gesamten Lebenszyklus hinweg optimiert werden. Diese Rollen arbeiten eng mit den Stakeholdern zusammen und sind häufig in Entscheidungsprozesse eingebunden.

Modellorientierte Rollen hingegen, darunter der MBSE-Anforderungsanalyst, der MBSE-Systemdesigner oder der API-Manager, fokussieren sich auf die strukturierte Modellierung von Systemen und deren Integration in digitale Entwicklungsumgebungen. Sie stellen sicher, dass Modelle konsistent, interoperabel und nachvollziehbar sind. Prozessorientierte Rollen fokussieren sich auf Umsetzung und Steuerung, während modellorientierte Rollen methodische und digitale Unterstützung leisten.

Beide Rollenarten sind eng miteinander verzahnt: Modellorientierte Rollen liefern strukturierte, analytische Grundlagen, die von prozessorientierten Rollen genutzt werden, um fundierte Entscheidungen zu treffen. Umgekehrt beeinflussen Anforderungen aus den Prozessen die Modellierungsmethoden und -inhalte.

In **Tabelle 4** sind die Teilnahmen und schwerpunktmäßigen Teilnahmen der Rollen an den technischen Prozessen aufgeführt.

Tabelle 4. Teilnahme modell- und prozessorientierter Rollen an technischen SE-Prozessen.

x = Teilnahme

xxx = Schwerpunktmäßige Teilnahme

		Technische Prozesse nach INCOSE Systems Engineering Handbuch													
		Geschäfts- und Missionsanalyse	Stakeholderbedarfe und Anforderungen	Systemanforderung	Architektur-gestaltung	Entwurf	Systemanalyse	Umsetzung	Integration	Verifizierung	Übergabe	Validation	Betrieb	Wartung	Entsorgung
Prozessorientierte Rollen	Anforderungsingenieur	xxx	xxx	xxx	x	x				x		x	x	x	x
	Verifizierungs- & Validationsingenieur	x	xxx	xxx						xxx		xxx			
	Systemarchitekt		x		xxx	xxx									
	Systemschnittstellenmanager		x	x	x	x			x						
	Technischer Manager			x	xxx	xxx	x	x	x	x					
	Leiter	xxx	x	x	x	x	x	x	x	x	xxx	x			
	Stakeholder-Schnittstellenmanager	x	xxx	x						x	xxx	x			
	Lebenszyklusingenieur	x	x	x	x		x	xxx	x	x	x	x	x	xxx	xxx
	Implementierungsingenieur				x	x		xxx	xxx	x					
	Informationsmanager	x	x	x	x	x	x	x	x	xxx	x	x	x	x	x
	Konfigurationsmanager				x	x	x	xxx	x	xxx	x	x	x	x	
	Prozesseigner	x	x	x	x	x	x	x	x	x	x	x	x	x	x
	Produktionsmanager					x		xxx	x	x					
Modellorientierte Rollen	Supportmanager	x	x	x		x	x						xxx	xxx	xxx
	MBSE-Anforderungsanalyst	xxx	xxx	xxx	x	x				x		x	x	x	x
	MBSE-Funktionsanalyst		x	xxx	x	x	xxx			x					
	MBSE-Systemdesigner		x	x	xxx	xxx	x	x	x	x					
	MBSE-Viewpointmanager	x	x	xxx	x	x	xxx	x	x						
	CAD-Ingenieur				x	xxx		xxx	xxx	x					
	MBSE-Systemanalyst			x	x	x	xxx			xxx		x	x	x	x
	CAM-Ingenieur			x				xxx	x	x					
	CAE-Ingenieur					x	xxx			xxx		x	x		
	Modellkoordinator	x	x	x	x	x	xxx			x	xxx	x			
	Modellstrangleiter			x	x	xxx	xxx	x	x	xxx	x	x	x		
	Interdisziplinärer Modellingenieur			x	x	x	xxx	x	x	x		x	x	x	x
	API-Manager		x	x	x	x	x	x		x		x	x	x	x

Die Bewertungen entstammen aus einer Analyse des IncoSE SE Handbooks [4], der Rollendefinition von Sheard [23], Gräßler et al. [25, 26], sowie aus dem Mapping von Kompetenzen aus dem Systems Engineering Competencies Framework (SECF)

[21] mit den technischen Prozessen des SE von IncoSE [4] im Systems Engineering Competency Assessment Guide [22].

In der Anforderungsanalyse arbeiten Anforderungsingenieure, Stakeholder-Schnittstellenmanager und MBSE-Anforderungsana-

lysten gemeinsam daran, Anforderungen systematisch zu erfassen und zu dokumentieren. Während die prozessorientierten Rollen sich auf die direkte Abstimmung mit Stakeholdern konzentrieren, stellen die modellorientierten Rollen sicher, dass die Anforderungen formalisiert und in strukturierte Modelle überführt werden.

Die Systemarchitektur und der Entwurf erfordern eine enge Abstimmung zwischen Systemarchitekten, Technischem Manager, MBSE-Systemdesignern und CAD-Ingenieuren. Dabei definieren prozessorientierte Rollen die funktionalen, strukturellen Anforderungen sowie den Vorentwurf, während modellorientierte Rollen die Anforderungen in digitale Modelle überführen und den deren Erfüllung durch den digitalen Entwurf erarbeiten.

Während der Systemanalyse und Umsetzung sorgen der MBSE-Systemanalyst, der CAD-Ingenieur und der CAE-Ingenieur für simulationsgestützte Analysen und Validierungen, während der Implementierungsingenieur, CAM-Ingenieur und der Produktionsmanager die praktische Umsetzung planen und überwachen. Die Interoperabilität zwischen verschiedensten Modellen wird hier durch den interdisziplinären Modellingenieur unterstützt, durch den Modellstrangleiter verantwortet und in der Ausprägung vom Modellkoordinator festgelegt.

In den Phasen der Integration, Verifizierung und Validierung übernehmen der Verifizierungs- & Validationsingenieur, der Konfigurationsmanager und der MBSE-Systemanalyst zentrale Aufgaben. Modellorientierte Rollen liefern hier simulationsbasierte Verifikationsansätze, während prozessorientierte Rollen praktische Tests und Prüfverfahren koordinieren.

Die Übergabe, der Betrieb und die Wartung erfordern die enge Zusammenarbeit von Lebenszyklusingenieuren, Supportmanagern und Informationsmanagern, um sicherzustellen, dass Systeme planmäßig genutzt und gewartet werden können. Modellorientierte Rollen spielen hier eine unterstützende Rolle, indem sie digitale Zwillinge und simulationsbasierte Wartungskonzepte bereitstellen.

In der Entsorgung sind besonders der Lebenszyklusingenieur und der Prozesseigner gefordert, nachhaltige Rückbau- und Recyclingstrategien zu entwickeln. Modellorientierte Rollen tragen dazu geringfügig bei, indem sie etwa Material- und Umweltauswirkungen digital analysieren.

Prozessorientierte und modellorientierte Rollen haben unterschiedliche, aber sich ergänzende Aufgabenbereiche im SE. Während prozessorientierte Rollen die Steuerung und Umsetzung der Prozesse übernehmen, sorgen modellorientierte Rollen für eine methodisch fundierte und konsistente Modellierung sowie digitale Umsetzung und Absicherung. Ihre enge Zusammenarbeit stellt sicher, dass Systeme effizient entwickelt, validiert und über ihren gesamten Lebenszyklus hinweg optimiert werden können. Dabei kann sowohl eine Person mehrere Rollen einnehmen als auch eine Rolle von mehreren Personen eingenommen werden. Ob mehrere Rollen von einer Person übernommen werden oder umgekehrt eine Rolle auf mehrere Personen verteilt wird, hängt stark von Projektgröße, Systemart und Komplexität ab. Dabei gilt es auch, die hinreichende Systemabsicherung und Expertise sowie Auslastung von Mitarbeitern gegenüber Aufwänden abzuwägen. Für kleinere Systementwicklungen kann es zum Beispiel sinnvoll sein den Anforderungsingenieur und den MBSE-Anforderungsanalysten einem Mitarbeiter zuzuordnen.

Alle genannten Rollen sind generisch sowie nicht produktspezifisch und unterstützen die erfolgreiche Umsetzung des System Engineerings mittels MBSE für die Entwicklung komplexer

Systeme. Diese Rollen bilden ein Basismodell, welches flexibel in verschiedenen Produktentwicklungsprozessen eingesetzt werden kann. Abhängig von der Systemart und der Unternehmens- und Projektgröße kann es sinnvoll sein, zusätzliche Rollen einzuführen, Rollen anzupassen oder Rollen aufgrund fehlender Anwendbarkeit nicht zu nutzen. Auch eine Kombination mit einem Teil des Rollenmodells und ein Weiterführen von Prozessanteilen ohne Verwendung des Rollenmodells zum Schutz etablierter und erfolgreicher Prozesse sowie weitere Individuallösungsarten sind denkbar.

4 Diskussion

Der vorgestellte Ansatz zur Systematisierung der Rollenzuteilung im SE bietet eine systematische Definition prozess- und modellorientierter Rollen. Er stellt einen Referenzrahmen bereit, auf dessen Grundlage Rollen in der Praxis strukturiert implementiert werden können. In Bezug auf Systemarten und Unternehmensgröße sind die Rollen jedoch nicht gleichermaßen anwendbar. Für anwendungsabhängige Abweichungen wird in diesem Ansatz kein Vorgehensschema bereitgestellt.

Die Differenzierung von Rollen zielt darauf ab, Zuständigkeiten im SE klarer zu strukturieren und operative sowie modellbasierte Aufgaben gezielt voneinander abzugrenzen. Insbesondere durch die Einführung spezialisierter modellorientierter Rollen sollen MBSE-bezogene Tätigkeiten entlastend aus dem Aufgabenbereich des generalistisch ausgerichteten Systems Engineers ausgelagert werden. Dies kann dazu beitragen, unkontrollierte Schnittstellen zu reduzieren, Verantwortlichkeiten zu schärfen und redundante Abstimmungen in interdisziplinären Entwicklungsteams zu vermeiden. Die erwarteten Effizienzpotenziale liegen insbesondere in einer verbesserten Anschlussfähigkeit von Modellen an reale Entwicklungsprozesse, einer beschleunigten Identifikation von Inkonsistenzen sowie in einer strukturierteren Nutzung digitaler Werkzeuge. Die strukturierte Koordination technischer Schnittstellen ist dabei ein zentrales Element.

Ein weiterer Vorteil liegt in der Möglichkeit, modellbasierte Informationen gezielter in die Entscheidungsebene der Prozessverantwortlichen zu überführen. Dadurch können fundierte Systementscheidungen beschleunigt und besser abgesichert werden. Dem stehen jedoch auch konkrete Herausforderungen gegenüber. Die Einführung neuer Rollen erfordert organisatorische Anpassungen, was vor allem in etablierten Unternehmensstrukturen auf Widerstand stoßen kann. Zudem steigt der initiale Schulungsaufwand, da modellorientierte Rollen spezifisches Tool- und Methodenwissen benötigen. Auch kann es zu einer temporären Fragmentierung kommen, wenn Rollen noch nicht eingespielt sind oder Überschneidungen unklar bleiben. Vor allem kleinere Unternehmen müssen hier Aufwand und Nutzen sorgfältig abwägen.

Das vorgestellte Rollenmodell knüpft bewusst an bestehende theoretische Fundamente an. So orientiert es sich an der Rollenstruktur von Sheard [23], dem erweiterten RM-SEA-Ansatz von Gräßler et al. [25] sowie den technischen Prozessen und Kompetenzverteilungen des IncoSE Systems Engineering Handbook [4]. Diese Anbindung gewährleistet, dass die Rollendifferenzierung nicht isoliert entwickelt wurde, sondern mit gängigen Strukturen und bestehenden Best Practices kompatibel ist. Eine konkrete Effizienzsteigerung in der industriellen Anwendung lässt sich zum jetzigen Zeitpunkt noch nicht empirisch belegen, erscheint jedoch angesichts der skizzierten strukturellen Entlastungspoten-

ziale und der theoretischen Fundierung plausibel. Voraussetzung ist eine projektspezifische Umsetzung, die durch entsprechende Qualifizierungsmaßnahmen begleitet wird.

Trotz theoretischer Fundierung und struktureller Anschlussfähigkeit bleibt eine zentrale Herausforderung die unternehmensspezifische Umsetzung. Die erfolgreiche Integration der Rollen setzt nicht nur strukturelle Anpassungen, sondern auch gezielte Weiterbildungsmaßnahmen voraus, insbesondere für MBSE-spezifische Methoden und Tools. Belastbare empirische Daten zum Kosten-Nutzen-Verhältnis liegen bislang nicht vor.

Gelingt die Umsetzung, kann das Rollenmodell langfristig zur Weiterentwicklung des MBSE beitragen, vor allem durch die strukturierte Entlastung des Systems Engineers. Eine stärkere Vernetzung von Modellierungsansätzen mit klassischen Entwicklungsprozessen fördert eine optimierte Integration von bestehenden Entwicklungsprozessen in Modellumgebungen. Dabei bieten insbesondere durchgängig vernetzte Werkzeuge Potenzial für eine verbesserte Absicherung und beschleunigte Abläufe. Dazu kann der API-Manager einen entscheidenden Beitrag leisten, auch unabhängig von bestehenden Vendorlösungen.

Insgesamt kann der vorgestellte Ansatz durch die Trennung von prozessorientierten und modellorientierten Aufgabenbereichen zur Effizienzsteigerung in Entwicklungsprozessen beitragen. Trotz Herausforderungen in der Implementierung bietet das Konzept großes Potenzial für die Weiterentwicklung des SE und die Optimierung komplexer technischer Systeme.

5 Schlussfolgerung

Die Differenzierung zwischen prozessorientierten und modellorientierten Rollen soll zur Effizienzsteigerung im SE beitragen. Sie verbessert die Interoperabilität zwischen digitalen Modellen und realen Entwicklungsprozessen. Gleichzeitig stellen die Implementierung in bestehende Unternehmensstrukturen sowie die notwendigen Qualifikationen der Fachkräfte Herausforderungen dar, die berücksichtigt werden müssen.

Der vorgestellte Ansatz verfügt über eine strukturierte Rollendefinition, die bestehende Kompetenzrahmen und Best Practices integriert. Diese Systematisierung der Rollen soll eine Referenz bieten, um die prozessseitige Anbindung von Modellierungen gezielt optimieren zu können und die Absicherung der technischen Prozesse und entsprechend der Systeme zu erweitern. Dazu sind gegebenenfalls Schulungen für Tools sowie strukturelle Anpassungen nötig, deren Aufwand und Nutzen sorgfältig abzuwägen sind. Die Umsetzung des Ansatzes kann langfristig zur Ausschöpfung der Potenziale des MBSE beitragen.

Die zukünftige Forschung sollte sich mit der praktischen Evaluation der Rollenzuteilung befassen, um empirisch basierte Weiterentwicklungen des Rollenmodells vornehmen zu können. Hierfür sind isolierte partielle Implementierungen denkbar, auf deren Basis Abstraktionen etwa zum Kosten-Nutzen-Verhältnis vorgenommen werden können. Darüber hinaus ist eine vertiefende Analyse der Kommunikation zwischen den Rollen notwendig, um sowohl generalisierbare Informationsflüsse als auch fallbezogene Limitationen zu identifizieren.

LITERATUR

- [1] Tschirner, C.: Rahmenwerk zur Integration des modellbasierten Systems Engineering in die Produktentstehung mechatronischer Systeme. Dissertation, Universität Paderborn, 2016
- [2] Fresemann, C.; Stark, R.; Sauer, C. et al.: Lücken und Herausforderungen bei der praktischen Umsetzung des Model-Based Systems Engineering. Konstruktion (2019) 06, S. 81–83, doi: [10.37544/0720-5953-2019-06-81](https://doi.org/10.37544/0720-5953-2019-06-81)
- [3] Thommes, K.; Iseke, A.; Schneider, M.: Digitales und prädiktives Kompetenzmanagement. Heidelberg: Springer Vieweg 2024
- [4] Incose: Incose Systems Engineering Handbook. Hoboken, New Jersey/USA: Wiley 2023
- [5] Verein Deutscher Ingenieure (VDI): VDI/VDE 2206 Entwicklung mechatronischer und cyber-physischer Systeme. Düsseldorf: VDI-Verlag 2021
- [6] Schlüter, N.: Generic Systems Engineering. Heidelberg: Springer Vieweg 2023
- [7] Albers, A.; Dumitrescu, R.; Gausemeier, J. et al. (Hrsg.): Strategie Advanced Systems Engineering – Leitinitiative zur Zukunft des Engineering und Innovationsstandorts Deutschland. Stand: 2022. Internet: www.advanced-systems-engineering.de/wp-content/uploads/ASE_Strategie.pdf. Zugriff am 08.07.2025
- [8] Rusch, F. R.; Willems, W.; Demke, N. et al.: Linking product development's and society's view on sustainability to enhance the contextual derivation and validation of requirements. IFAC-PapersOnLine 58 (2024) 19, pp. 1198–1203
- [9] Schneider, B.; Schüle, S.; Kürümlüoğlu, M. et al.: Advanced Systems Engineering unternehmensindividuell konfigurieren und einführen – Advanced Systems Engineering Assessment. WT Werkstattstechnik 114 (2024) 6, S. 259–267. Internet: www.werkstattstechnik.de. Düsseldorf: VDI Fachmedien
- [10] Douglass, B. P.: Agile Systems Engineering. Amsterdam: Elsevier Inc. 2016
- [11] Rusch, F. R.; Demke, N.; Willems, W. et al.: Context-based Derivation of Holistic Sustainability Requirements in the Early Phase of Product Development. Procedia CIRP 122 (2024), pp. 306–311
- [12] Baughey, K.: Functional and Logical Structures: A Systems Engineering Approach. SAE Technical Paper (2011) 2011–01–0517, doi: [10.4271/2011-01-0517](https://doi.org/10.4271/2011-01-0517)
- [13] Hoffmann, H.-P.: Systems Engineering Best Practices with the Rational Solution for Systems and Software Engineering. Somers NY: Deskbook Release 4.1 2013
- [14] Péraire, C.; Edwards, M.; Fernandes, A. et al.: The IBM Rational Unified Process for System z. Stand: 2007. Internet: www.redbooks.ibm.com/redbooks/pdfs/sg247362.pdf. Zugriff am 08.07.2025
- [15] Voirin, J.-L.: Model-based System and Architecture Engineering with the Arcadia Method. London: ISTE Press – Elsevier 2017
- [16] Salehi, V.; Florian, G.; Taha, J.: Implementation of Systems Modeling Language (SysML) in Consideration of the CONSENS Approach. DS 92: Proceedings of the DESIGN 2018 15th International Design Conference, 2018, pp. 2987–2998
- [17] Pearce, P.; Hause, M.: ISO-15288: OOSEM and Model-Based Submarine Design. Stand: 2012. Internet: www.omg.org/sysml/Pearce_Hause_ISO-15288_OOSEM_and_Model-Based_Submarine_Design_SETE_APCOSE_20121.pdf. Zugriff am 08.07.2025
- [18] London, B.: A Model-Based Systems Engineering Framework for Concept Development. Cambridge, MA: Massachusetts Institute of Technology 2012
- [19] Robinson, K.; Tramondans, D.; Harvey, D. et al.: Demonstrating Model-Based Systems Engineering. Systems Engineering / Test & Evaluation Conference, Adelaide, 2010. Internet: www.shoalgroup.com/wp-content/uploads/2017/06/Robinson-et-al-2010-Demonstrating-Model-Based-Systems-Engineering-for-Specifying-Complex-Capability-SETE-2010.pdf. Zugriff am 08.07.2025
- [20] Estefan, J.: Survey of Model-Based Systems Engineering (MBSE) Methodologies. Incose MBSE Initiative, Pasadena, CA, 2008. Internet: https://www.omg.org/sysml/MBSE_Methodology_Survey_RevB.pdf. Zugriff am 08.07.2025
- [21] Incose: Incose Systems Engineering Competency Framework. Incose Technical Product Reference: Incose-TP-2018-002-01.0. Stand: 2018. Internet: www.incose.org/docs/default-source/professional-development-portal/isecef.pdf?sfvrsn=dad06bc7_4. Zugriff am 08.07.2025

- [22] Presland, I.; Whitcomb, C.; Zipes, L. (eds.): ECF Annex D: IncoSE Systems Engineering Competency Framework. Systems Engineering Competency Assessment Guide. Hoboken, New Jersey/ USA: John Wiley & Sons Inc. 2023
- [23] Sheard, S.: Twelve Systems Engineering Roles. Systems Engineering Practices & Tools 6 (1996) 1, pp. 478–485
- [24] Sheard, S.: The Value of Twelve Systems Engineering Roles. Systems Engineering Practices & Tools 6 (1996) 1, pp. 894–902
- [25] Gräßler, I.; Oleff, C.; Hentze, J.: Role Model for Systems Engineering Application. Proceedings of the Design Society: International Conference on Engineering Design, 2019, pp. 1265–1274, doi.org/10.1017/dsi.2019.132
- [26] Gräßler, I.; Thiele, H.; Grewe, B. et al.: Responsibility Assignment in Systems Engineering. Proceedings of the Design Society 2 (2022) 1, pp. 1875–1884, doi.org/10.1017/pds.2022.190
- [27] Hage, H.: Rahmenwerk zur automatisierten Verifikation eines systemorientierten Produktentstehungsprozesses in der Automobilentwicklung. Dissertation, Helmut-Schmidt-Universität Hamburg, 2024

Fabian Rusch, M.Sc. 
fabian.rusch@hsu-hh.de

Dr.-Ing. Hassan Hage

Prof. Dr.-Ing. Frank Mantwill

Helmut-Schmidt-Universität Hamburg
 Holstenhofweg 85, 22043 Hamburg
www.hsu-hh.de/mrp/

LIZENZ



Dieser Fachaufsatz steht unter der Lizenz Creative Commons
 Namensnennung 4.0 International (CC BY 4.0)