

Self-Enforcing Networks als Tools zur Auswahl eines geeigneten (ggf. hybriden) Vorgehensmodells in IT-Projekten

Christina Klüver¹ und Jürgen Klüver²

Abstract: Die Auswahl eines geeigneten Vorgehensmodells (VM) für bestimmte Projekte oder als institutionelle Vorgabe erfordert nicht nur die Kenntnis der zahlreichen VM sondern auch die Zuordnung zu den Anforderungen des jeweiligen Projekts. Unser Artikel beschreibt die Möglichkeiten, die für diese Probleme ein von uns entwickeltes selbstorganisiert lernendes neuronales Netz bietet. Dies Netz, das Self Enforcing Network SEN, wird in seiner Grundlogik beschrieben; seine praktischen Verwendungsmöglichkeiten werden in verschiedenen Szenarien demonstriert. Die wesentlichen praktischen Vorteile von SEN sind Flexibilität sowie leichte Bedienbarkeit und Interpretation der Ergebnisse.

Keywords: Self-Enforcing Network, Selbstorganisiertes Lernen, Vorgehensmodelle

1 Einleitung

Die Auswahl eines geeigneten Vorgehensmodells stellt die Verantwortlichen bekanntlich vor große Herausforderungen: Die Vorgehensmodelle müssen generell bekannt sein, die Vorgehensweise in IT-Projekten weisen häufig besondere Charakteristika auf, die Kunden haben ihrerseits bestimmte Vorstellungen, die nicht unbedingt mit dem Auftrag kompatibel sind, und für Zertifizierungen müssen bestimmte Vorgaben eingehalten werden – um nur einige der wesentlichen Probleme zu nennen. Erschwerend kommt noch dazu, dass es seit einiger Zeit eine Vielfalt von Vorgehensmodellen gibt (u.a. [KL14]), die für verantwortliche Manager nur noch schwer zu durchschauen ist. Eine Unterstützung der entsprechenden Auswahlprozesse durch geeignete Softwaretools wäre demnach für die betriebliche Praxis eine wesentliche Hilfe.

Bekannt sind mittlerweile unterschiedliche Klassifizierungen (u.a. [WM10]) wie insbesondere konzeptionelle, sequentielle, inkrementelle, iterative, spiralförmige und agile Vorgehensmodelle, die sich im Laufe der Jahre durchgesetzt haben.

Wir stellen in diesem Artikel ein Softwaretool vor, das Entscheidungsunterstützungen bei der Auswahl von Vorgehensmodellen ermöglicht, die für ein neues Projekt am besten geeignet sind. Das Tool basiert auf einem von uns entwickelten künstlichen neuronalen

¹ Universität Duisburg-Essen, Fakultät WiWi, Universitätsstr. 12, 45117 Essen, christina.kluever@uni-due.de

² Universität Duisburg-Essen, COBASC-Research Group, Universitätsstr. 12, 45117 Essen, juegen.kluever@uni-due.de

len Netz, das auf der Basis des selbstorganisierten Lernens operiert – das Self Enforcing Network (SEN). Das Tool kann flexibel gemäß unterschiedlichen Bedürfnissen eines Benutzers eingestellt werden und liefert in einem Visualisierungsteil anschaulich die Resultate als Empfehlungen für die Benutzer. Es zeigt nach Bedarf auch, welche unterschiedlichen Vorgehensmodelle kombinierbar sind.

Die folgende Sektion beschreibt das methodische Vorgehen, um eine geeignete Datenbasis für das SEN zur Verfügung zu stellen. Im dritten Kapitel wird das Self Enforcing Network näher dargestellt; im vierten Kapitel wird die Operationsweise des SEN und die Visualisierung der Resultate an einigen praktischen Beispielen demonstriert. Wünschenswerte Erweiterungen des Tools und zusätzliche praktische Möglichkeiten werden am Schluss skizziert.

2 Der methodische Ansatz zur Auswahl von Vorgehensmodellen

Der hier dargestellte Prototyp des SEN enthält vierzehn Vorgehensmodelle – klassische wie agile -, wie z.B. „Wasserfall“, „Scrum“ und „RUP“. Diesen werden insgesamt sechzig Attribute zugewiesen, die als Kriterien bzw. als Charakteristika der jeweiligen Vorgehensmodelle fungieren. Um eine Zuordnung der entsprechenden Attribute zu den Modellen transparenter zu gestalten, werden die Kriterien in zwölf Gruppen aufgeteilt, die unterschiedliche Aspekte berücksichtigen. In Tabelle 1 werden die Gruppen mit einigen der Aspekte aufgeführt:

Gruppe	Anzahl Kriterien	Beispiele
Ökonomische Kriterien	6	Budget, Projektdauer, (Schulungs-)Aufwand...
Flexibilität	6	Einsatz, Benchmark-Fähigkeit, Unterstützung durch Benchmarksysteme (CMMI, ITIL...)
Projektcharakter	7	Erfordert Kreativität, diplomatischer Umgang im Projekt, Anzahl der Arbeitspakete
Umwelt	2	Beachtung neuer Entwicklungen, Änderungen innerhalb des Projektverlaufs
Dokumentation	3	Detailierungsgrad, Häufigkeit, Beteiligte
Risikomanagement	9	Verzögerung, Kundenkontakt, Änderung des Codes
Team	4	Größe, Erfahrungsgrad, Anzahl der MA
Kommunikation	5	Häufigkeit, Ort, Kommunikationswege
Akzeptanz	4	Offenheit für neue Methoden, Rollenmodell, Hierarchie-Support
Phasenbindung	6	Qualitätsmanagement, Methodenbindung, festgelegte Prozesse
Technik	4	Herstellerabhängigkeit, Entwurfsobjektbindung
Formalisierung	4	Formalisierungsgrad, QS-Unterstützung, Zertifizierungssupport

Tab. 1: Beispiele für die Zuordnung der Kriterien

Die Auswahl der Kriterien entstand auf der Basis von Literaturrecherche und durch Diskussionen mit Managern bzw. Mitarbeitern verschiedener Unternehmen, die mit unterschiedlichen VM vertraut sind und über praktische Erfahrungen mit mindestens einem VM verfügen.

Zunächst wurde für jedes VM eine Liste mit den wesentlichen Charakteristika erstellt. Anschließend wurden die Listen zusammengeführt und Doppelungen eliminiert, so dass alle Kriterien nur einmal aufgeführt werden.

Im nächsten Schritt erfolgten die Bewertungen der Charakteristika für alle VM mit Werten zwischen 0.0 und 1.0; 1.0 bedeutet, dass das Charakteristikum für das VM sehr wichtig ist, 0.0 natürlich, dass es völlig unwesentlich ist. Beispielsweise wird die "*Dokumentation*" hinsichtlich des Detaillierungsgrads bei RUP mit 1.0 bewertet, hingegen bei Build & Fix mit 0.0. Bei dem Kriterium "*Monetäres Budget*" wird angenommen, dass Unternehmen, die das V-Modell XT, das Spiral- oder Wasserfallmodell verwenden, über ein sehr hohes Budget verfügen können; in diesem Fall steht eine 1.0 wenn das Budget mehr als 1.000.000 € betragen kann, entsprechend eine 0,1 wenn es sich um ein Projekt handelt, das maximal über 5.000 € verfügt. "*Projektdauer*" wird mit 0.3 bewertet, wenn es sich um Projekte handelt, die maximal für 30 Tage geplant werden und für die als VM eher XP oder Srum in Frage kommen. Bei dem V-Modell XT oder Wasserfall Modell wird die Dauer mit 1.0 bewertet, da die Projekte mit einer Laufzeit von mehr als fünf Jahren vorgesehen werden können.

Auf dieser Weise wurde eine Matrix entwickelt mit vierzehn Zeilen für die VM als Objekte und sechzig Spalten mit den bewerteten Attributen, die dem Netzwerk 1:1 übergeben wird (s. Abb. 1).

3 Das Self-Enforcing Network (SEN)

Beim SEN handelt es sich um ein künstliches neuronales Netzwerk, das durch selbstorganisiertes (bzw. nicht überwachtes) Lernen charakterisiert ist. Ein SEN besteht aus drei wesentlichen Komponenten: einer semantischen Matrix, dem eigentlichen Netzwerk und der Visualisierungskomponente (für detailliertere Beschreibungen sowie verschiedene Anwendungsbeispiele vgl. z.B. [KKS12] sowie [KK13]).

In der *semantischen Matrix* sind die „semantischen“ Beziehungen zwischen Objekten und Attributen enthalten, die vom SEN geordnet werden sollen; hier geht es um die Beziehung zwischen Vorgehensmodellen und deren Charakteristika. Die semantische Matrix kann zum einen binär codiert werden, wenn ein Attribut zu einem Objekt vorliegt (1) oder nicht vorliegt (0), oder zum anderen reell codiert, wenn man die relative Zugehörigkeit des Attributs zu einem Objekt ausdrücken möchte (stark, mittel, schwach usw.). In Abb. 1 wird ein Ausschnitt gezeigt:

Objekt Name	Monetäres Budget	Anzahl Teilprojekte	Schulungsaufwand	Implementierungsaufwa...
XP	0,50	0,10	0,20	0,10
SCRUM	0,30	0,40	0,30	0,20
RUP	0,80	0,90	0,80	0,80
V-Modell 97	1,00	0,70	0,50	0,50
V-Modell XT	1,00	1,00	0,80	0,70
Wasserfall Modell	1,00	0,70	0,70	0,40
Spiral Modell	1,00	0,80	0,70	0,60
Chrysal	0,50	0,60	0,50	0,50
ASD	0,50	0,50	0,50	0,20
FDD	0,50	0,50	0,50	0,60
DSM	0,70	0,30	1,00	0,60
LSD	0,70	0,50	0,60	0,30
MDD	0,50	0,50	0,70	0,60

Abb. 1: Ausschnitt aus der semantischen Matrix

Aus der semantischen Matrix wird das eigentliche *Netzwerk* des SEN konstruiert. Dies geschieht durch die Generierung einer Gewichtsmatrix, in der den Verbindungen zwischen den Objekten und den Attributen Gewichte zugeordnet werden. Die Objekte und die Attribute stellen Neuronen dar, die miteinander voll verbunden sein können. Bei der Initialisierung der Gewichtsmatrix sind alle Gewichte $w = 0$. Aus der semantischen Matrix werden mit folgender Lernregel die Gewichtswerte generiert:

$$(t + 1) = w(t) + \Delta w \quad (1)$$

$$\Delta w = c * v_{sm}$$

Hier ist c eine Lernrate; diese wird gewöhnlich als $c = 0.1$ gesetzt. Bei v_{sm} handelt es sich um den entsprechenden Wert aus der semantischen Matrix, also dem Beziehungswert zwischen einem Objekt und einem Attribut.

Die Dynamik eines SEN wird wie üblich bei neuronalen Netzen durch eine spezifische Aktivierungsfunktion generiert; das SEN bietet hier sechs Möglichkeiten zur Auswahl an. Bei praktischen Anwendungen reicht es häufig, die einfache *lineare Aktivierungsfunktion* (a_j) zu verwenden:

$$a_j = \sum w_{ij} * a_i \quad (2)$$

Bei gewünschten Dämpfungen der Ergebnisse steht die von uns entwickelte *logarithmisch-lineare Aktivierungsfunktion* zur Verfügung:

$$a_j = \sum \begin{cases} \lg_3(a_i + 1) * w_{ij}, \\ \lg_3(|a_i + 1|) * -w_{ij} \end{cases} \quad (3)$$

Diese Funktion wurde vor allem deshalb entwickelt, da bei großen Netzen wegen der Linearität der einfachen linearen Funktion die Aktivierungswerte der Neuronen rasch sehr groß werden können. Die zusätzliche Einführung des Logarithmus bewirkt eine entsprechende Dämpfung dieser Werte.

Zusätzlich verfügbar sind weitere Varianten der linearen Aktivierungsfunktion, nämlich die sog. *lineare Aktivierungsfunktion mit Mittelwert* und die sog. *Enforcing Activation Function*, die ebenfalls von uns entwickelt wurden. Außerdem gibt es die Möglichkeiten, die *Tangens Hyperbolicus Funktion* sowie die bekannte logistische Aktivierungsfunktion zu verwenden [KKS12].

Die im nächsten Kapitel gezeigten Anwendungsbeispiele verwenden nur die einfache lineare Aktivierungsfunktion.

Für die Klassifikation eines oder mehrerer Objekte werden die entsprechenden Attributs-Neuronen extern aktiviert, die die Objekte charakterisieren. Die Endaktivierungswerte der Objektsneuronen entscheiden einerseits über die Ähnlichkeit zwischen verschiedenen Objekten (Vorgehensmodellen) und andererseits über die Eignung der verschiedenen Vorgehensmodelle für ein bestimmtes Projekt. „Externe Aktivierung“ bedeutet die Belegung eines Neurons mit einem bestimmten numerischen Wert – im folgenden Beispiel 1.0 ; anschließend berechnet das Netz über die Aktivierungsfunktion die Werte der Endaktivierungen der Objektneuronen. Die folgende Graphik demonstriert dies Verfahren; die externe Aktivierung wird sozusagen von den Attributsneuronen an die Objektneuronen in modifizierter Form durch die lineare Aktivierungsfunktion weiter gegeben:

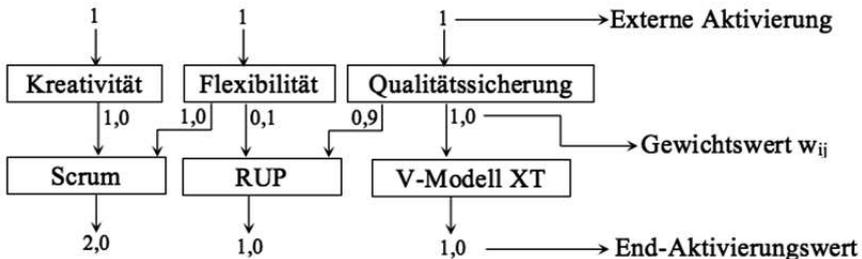


Abb. 2: Beispiel für die Aktivierung einzelner Neuronen. Dargestellt wird nur ein Teil der Verbindungen, um die Übersichtlichkeit zu gewährleisten

Aufgrund der Stärke und Anzahl der Verbindungen wird offenbar „Scrum“ am stärksten aktiviert.

Für die *Visualisierung* wird, wie bemerkt, die Stärke der Endaktivierung zwischen den Objektneuronen verglichen. Dabei sind Objekte mit annähernd gleichen Endaktivierungen ähnlich bzw. Objekte mit stark unterschiedlichen Endaktivierungen unähnlich. SEN visualisiert dies durch die Darstellung der Distanz zwischen den Objekten. Eine nahe Distanz entspricht einer hohen Ähnlichkeit und umgekehrt. Die Distanz zwischen den jeweiligen Objekten ist die Differenz ihrer jeweiligen Endaktivierungswerte.

SEN bietet drei verschiedene Möglichkeiten an, die Ergebnisse für einen Benutzer anschaulich darzustellen. Die erste Möglichkeit ist die von uns so genannte *Zentrumvisualisierung*. Diese dient vor allem dazu, dem Benutzer graphisch darzustellen, welche Vorgehensmodelle für sein Projekt am besten geeignet sind. Dazu wird das Projekt als Vektor mit den entsprechenden Anforderungen ins Zentrum eines zweidimensionalen Gitters platziert und durch die externe Aktivierung der Attributsneuronen werden dann die Symbole für die verschiedenen Vorgehensmodelle ins Zentrum des Gitters gezogen. Je näher ein Modellsymbol dem Zentrum ist, desto geeigneter ist das entsprechende Vorgehensmodell für das Projekt des Benutzers.

Die zweite Möglichkeit ist die sog. *Kartenvisualisierung*. Diese demonstriert die relative Ähnlichkeit der Objekte zueinander, indem die Objekte, hier die Vorgehensmodelle, zu Clustern geordnet werden. Daraus lässt sich entnehmen, welche Vorgehensmodelle ggf. füreinander substituierbar sind. Technisch wird dies, sehr einfach ausgedrückt, durch die Berechnung der Euklidischen Distanz realisiert: Je größer die Distanz zwischen zwei Vektoren der VM ist, desto weiter voneinander werden diese im Koordinatensystem platziert.

Die dritte Möglichkeit ist die sog. *Expertenansicht*. Bei dieser werden die Vorgehensmodelle tabellarisch danach geordnet, welche für ein Projekt am besten geeignet sind. Die Expertenansicht ist demnach eine andere Darstellungsform der Zentrumsvisualisierung. In den folgenden Beispielen werden alle drei Visualisierungsmöglichkeiten gezeigt.

4 Ergebnisse

Um die Potentiale der Auswahl mit einem SEN zu veranschaulichen werden unterschiedliche Szenarien aufgeführt.

4.1 Ordnung von Basismodellen in einschlägigen Clustern

Für eine Clusterordnung von Basismodellen gemäß der Option Kartenvisualisierung wurden lediglich einige der bekanntesten Modelle, nämlich Wasserfall-, Spiral-, V-Modell 97, RUP, V-Modell XT, Scrum und XP gewählt. Für alle Vorgehensmodelle (VM) wurden die Werte in der semantischen Matrix gemäß ihrer Ausprägung für das jeweilige VM definiert.

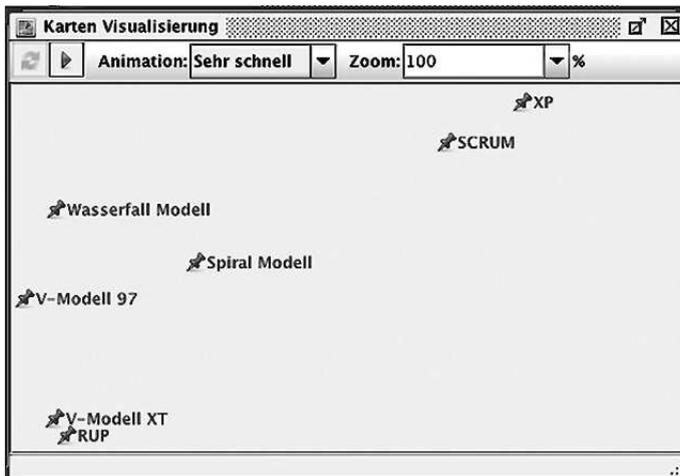


Abb. 3: Clustern der Basismodelle

Anhand der Platzierung der einzelnen VM auf der Visualisierungskarte sind drei Cluster erkennbar. Scrum und XP bilden gemeinsam einen Cluster; das Ergebnis ist nachvollziehbar, da beide für kleinere Projekte geeignet sind und insbesondere zu den bekanntesten agilen VM gehören. In der Softwareentwicklung wird entsprechend eine Kombination von Scrum mit XP als vielversprechend empfohlen [Hi09] und diese wurde in der Praxis bereits durchgeführt [DD08; DD14].

RUP und V-Modell XT bilden ebenfalls einen Cluster. Beide können als generische Metamodelle betrachtet werden, Risikomanagement sowie Qualitätssicherung werden berücksichtigt und die Vorgehensweise wird ganz genau geplant.

Die Clusterung am linken Rand ist insbesondere bei dem Wasserfall- und V-Modell 97 nicht überraschend, da letzteres eine Erweiterung des bekannten Wasserfallmodells ist.

Da mit dem Spiralmodell versucht wurde, andere Vorgehensmodelle als "Sonderfälle" zu integrieren [Ga08, 72] ist es ebenfalls plausibel, dass dieses VM eher zentriert platziert wird. Die Ergebnisse des SEN sind demnach durchaus kompatibel mit den bekannten Eigenschaften der verschiedenen VM und deren Ähnlichkeiten untereinander.

4.2 Erweiterung eines SEN durch weitere Vorgehensmodelle

Mittlerweile existiert eine große Anzahl an Vorgehensmodellen, die in der Praxis unterschiedlich eingesetzt werden [KL14]. Wenn man neue Vorgehensmodelle durch SEN einordnen lassen will, ist es ohne große Probleme möglich, die bisherige semantische Matrix durch die Attribute dieser neuen VM zu erweitern und die neuen VM entsprechend einer der Optionen ordnen bzw. sie auf ihre Eignung für ein bestimmtes Projekt überprüfen zu lassen.

Für sieben neue VM, die im bisher verwendeten SEN noch nicht enthalten waren, ergibt eine Clustering in der Kartenvisualisierung folgendes Bild:

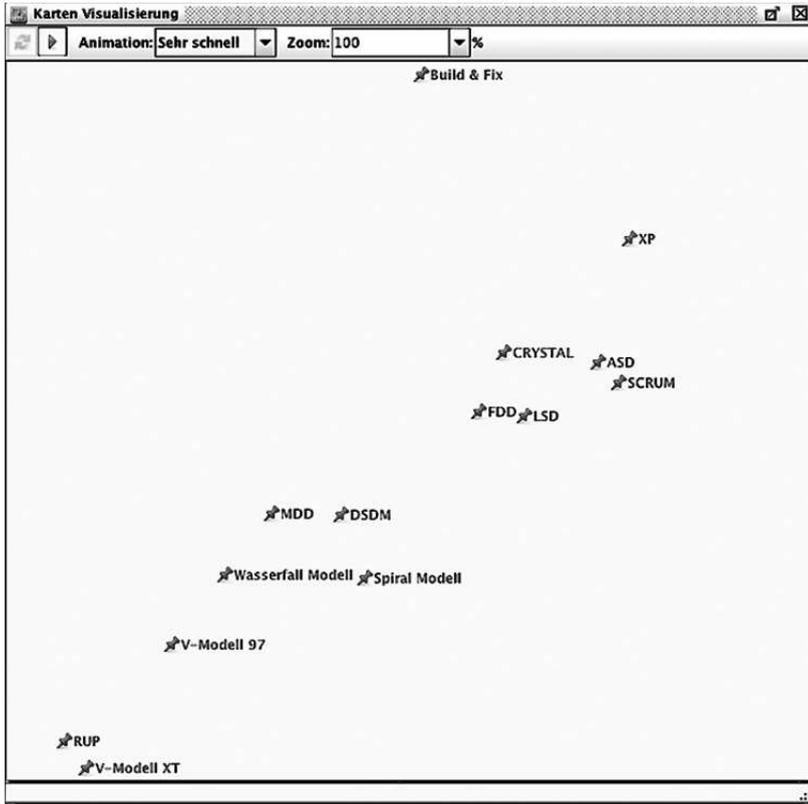


Abb. 4: Einordnung neuer VM

Die Erweiterung der semantischen Matrix durch die neuen Vorgehensmodelle führte durch das Lernverfahren zu einer leicht veränderten Ordnung der bereits implementierten Vorgehensmodelle; die grundsätzliche Ordnung ist jedoch erhalten geblieben.

In den letzten Jahren wurden insbesondere agile Vorgehensmodelle miteinander verglichen, um Vor- wie Nachteile, Gemeinsamkeiten und Unterschiede herauszuarbeiten [CH11; DD14; FC14]. Die Ergebnisse von SEN scheinen mit diesen Untersuchungen kompatibel zu sein.

4.3 Auswahl eines geeigneten Vorgehensmodells für ein Projekt

Damit eine konkrete Empfehlung durch das SEN erfolgen kann, muss ein Benutzer einen Vektor erstellen, einen sog. Eingabevektor, bestehend aus den Attributen, die das Projekt charakterisieren.

Als Beispiel wird unsere eigene Forschungsgruppe herangezogen und zwar in der Situation, wenn ein neuer Algorithmus implementiert werden soll. Erfahrungsgemäß ist in solchen Phasen der Code häufig zu verändern, da sich erst in der Entwicklung zeigt, ob der Algorithmus verfeinert werden muss. In dieser Phase wird selten dokumentiert, auch wenn die Entwicklung anders verlaufen sollte als geplant, die Implementierung und Funktionalität stehen dagegen im Vordergrund.

Folgende gewichtete Kriterien wurden demnach als Eingabevektor eingegeben:

	Kreativität	Code- Änderungen	Qualifikation MA	Teamgröße	Erfahrung der MA
Projekt Start	1.0	0.9	0.9	0.1	0.8

Das Ergebnis mit SEN sieht folgendermaßen aus:

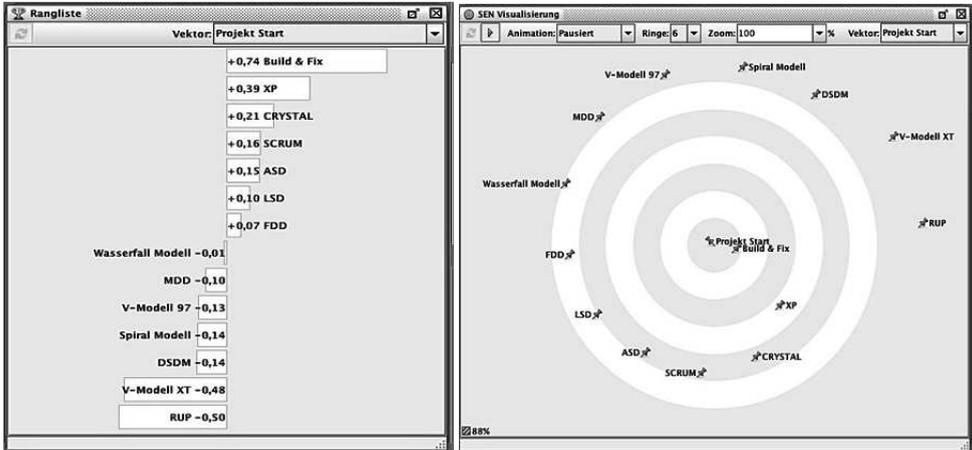


Abb. 5: Geeignete Vorgehensweise in der eigenen Forschungsgruppe, links ist die erwähnte Expertenansicht

Das Ergebnis zeigt, dass das Vorgehen eher Build & Fix entspricht, was zu Beginn eines Projektes auch zutrifft. Erst wenn der Algorithmus funktioniert und hinreichend getestet wurde, erfolgt eine Dokumentation.

Wie in Abb. 5 zu erkennen ist, werden auch negative Aktivierungswerte gezeigt. Das liegt darin, dass die Rohdaten im Wertebereich $[-1,1]$ abgebildet werden. Dieses Vorgehen bietet sich an, wenn die semantische Matrix ebenso viele Nullen wie Einsen enthält

(s. Abb. 1). Dadurch wird vermieden, dass nur Neuronen – in diesem Fall VM – am stärksten aktiviert werden, deren Attributswerte überwiegend hoch oder eins sind.

Gerade bei Forschungsprojekten ist es natürlich denkbar, dass nach Überprüfung des Projektstandes über Variationen des benutzen VM nachgedacht werden muss. Das ist durch Eingabe neuer Kriterien in SEN problemlos möglich.

4.4 Hybride Modelle

Unter hybriden Modellen verstehen wir hier, wie in der Einleitung erwähnt, die Kombination von klassischen mit agilen VM [BKS14]. Eine Unterstützung der Entscheidung, welche Kombinationen für bestimmte Projekte geeignet sind, kann durch SEN beispielsweise folgendermaßen geschehen: Wenn in einem Projekt bereits ein klassisches VM wie V-Modell XT eingesetzt wird, kann SEN zeigen, welche agilen VM am besten mit V-Modell XT kombiniert werden können. Ein entsprechendes Resultat zeigt Abb. 6:

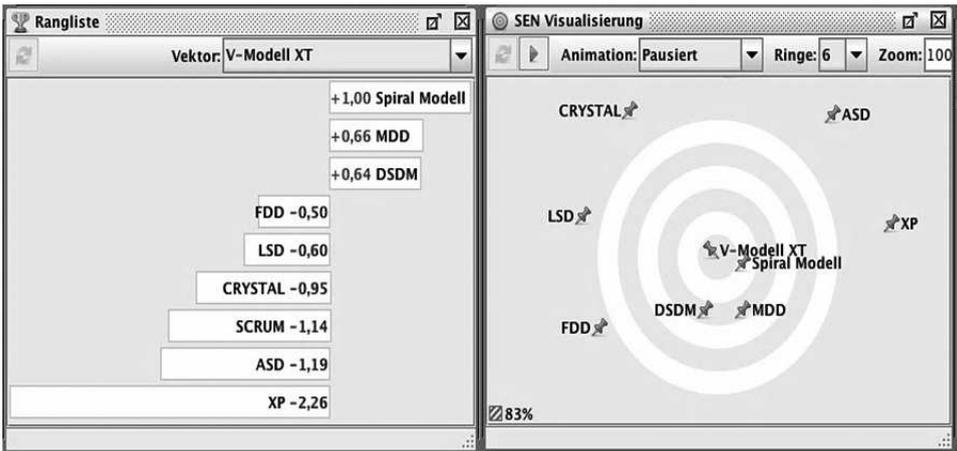


Abb. 6: Vereinbarkeit von V-Modell XT mit agilen VM

Offenbar sind für eine Kombination mit V-Modell XT das Spiralmodell und dann MDD sowie DSDM am besten geeignet.

5 Ausblick

SEN ist seit seiner Entwicklung mehrfach bei unterschiedlichen praktischen Problemen erfolgreich eingesetzt worden wie z.B. bei der Auswahl geeigneter Standorte für Offshore Windanlagen [KKS11], als Orientierungssystem für Studienanfänger und als Entscheidungsunterstützung für die Auswahl von Betriebsrichtungen am Flughafen Frank-

furt. Offenbar ist SEN auch sehr gut dafür geeignet, Entscheidungshilfe für Projektmanager bei der Auswahl von VM zu liefern. Ein Manager, der die Aufgabe hatte, ein geeignetes VM für eine IT-Firma und für bestimmte Projekte einzuführen, brauchte laut eigener Auskunft fünf Manntage, um eine Entscheidung zu treffen. Durch SEN wurde das ermittelte VM bestätigt und darüber hinaus wurden die nächst Platzierten als eine sinnvolle Alternative betrachtet. In dieser ersten Evaluierungsphase wurde das entwickelte SEN-Modell ebenfalls für die Meilensteinanalyse untersucht, um festzustellen, ob durch Projektänderungen andere VM geeigneter sind.

Einer der praktischen Vorteile von SEN ist die problemlose Erweiterung durch neue Vorgehensmodelle wie zum Beispiel mit Kanban, sowie die beliebig feine Differenzierung der Kriterien, nach denen die VM beurteilt werden. Darüber hinaus können Kriterien für das Tailoring [u.A. KaK13] geprüft und durch die Projektbeteiligten, je nach Projektart, bestimmt werden.

In naher Zukunft soll die Erstellung eines evaluierten semantischen Netzes durch Kooperation mit Experten und durch empirische Studien erfolgen.

Literaturverzeichnis

- [BKS14] Berg, B.; Knott, P.; Sandhaus, G.: *Hybride Softwareentwicklung: Das Beste aus klassischen und agilen Methoden in einem Modell vereint*. Berlin Heidelberg: Springer-Verlag, 2014.
- [CH11] Chowdhury, A.F.; Huda, M.N.: *Comparison between Adaptive Software Development and Feature Driven Development*. International Conference on Computer Science and Network Technology (ICCSNT), vol.1, S. 363-367, IEEE *Xplore*, 2011.
- [DD08] Dybå, T.; Dingsøy, T.: *Empirical studies of agile software development: a systematic review*. Information and Software Technology 50 (August (9/10)), S. 833-859, 2008.
- [DD14] Diebold, P.; Dahlem, M.: *Agile Practices in practice: a mapping studies*. In: EASE '14 Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering. New York: ACM, 2014.
- [FC14] Flora, H.K.; Chande, S.V.: *A Systematic Study on Agile Software Development Methodologies and Practices*. International Journal of Computer Science and Information Technologies (IJCSIT), Vol. 5 (3), S. 3626-3637, 2014.
- [Ga08] Gadatsch, A.: *Grundkurs IT-Projektcontrolling*. Wiesbaden: Vieweg+Teubner, 2008.
- [Hi09] Hindel, B. et al.: *Basiswissen Softwaremanagement*. Heidelberg: dpunkt Verlag, 2009.
- [KaK13] Kalus, G.; Kuhrmann, M.: *Criteria for software process tailoring: a systematic review*. In: ICSSP 2013 Proceedings of the 2013 International Conference on Software and System Process. New York: ACM, S. 171-180, 2013
- [KK11] Klüver, C.; Klüver, J.: *IT Management durch KI-Methoden und andere naturanaloge Verfahren*. Wiesbaden: Vieweg Teubner, 2011.

- [KKS12] Klüver, C.; Klüver, J., Schmidt, J.: Modellierung komplexer Prozesse durch naturanaloge Verfahren. Soft Computing und verwandte Techniken. 2. erweiterte Auflage. Wiesbaden: Springer Vieweg, 2012.
- [KK13] Klüver, C.; Klüver, J.: Self-Organized Learning by Self Enforcing Networks. In (Rojas, I., Joya, G., Cabestany J. Hrsg.): Advances in Computational Intelligence. Heidelberg Dordrecht London New York: Springer, 2013.
- [KL14] Kuhrmann, M.; Linssen, O.: Welche Vorgehensmodelle nutzt Deutschland? In (Engstler, M.; Hanser, E.; Mikusz, M.; Herzwurm, G. Hrsg.): Projektmanagement und Vorgehensmodelle 2014. Soziale Aspekte und Standardisierung Lecture Notes in Informatics (LNI) - Proceedings Series of the Gesellschaft für Informatik (GI) Volume P-236 Bonne Köllen Verlag, S. 17-33, 2014.
- [WM10] Wiczorrek, H.W.; Mertens, P.: Management von IT-Projekten: Von Der Planung Zur Realisierung. Springer, London, Dordrecht, Heidelberg, New York, 2010.