

Die Essenz des Software Engineering – spielerisch und integriert

Jöran Pieper, Institute for Applied Computer Science, FH Stralsund

Joeran.Pieper@fh-stralsund.de

Zusammenfassung

Softwareprozesse und Software Engineering Methoden gehören zu den Wissensgebieten des Software Engineering (SE), deren anschauliche Vermittlung besonders herausfordernd ist.

Mit der Spezifikation „*Essence – Kernel and Language for Software Engineering Methods*“ (Object Management Group, 2014) der *SEMAT Initiative* (SEMAT, 2014) existiert nun ein Ansatz, der u.a. verspricht, alle essentiellen Dimensionen von SE-Aufgaben in einem kompakten universellen und ausführbaren Kernel zusammenzufassen.

Dieser Beitrag beschreibt die Eigenschaften der *Essence*-Spezifikation im Hinblick auf Ihre Eignung, Studierende in die Welt der SE-Methoden einzuführen.

Um ein tiefes Verständnis der *Essence*-Konzepte zu ermöglichen, bedarf es eines geeigneten Ansatzes. Der hier vorgestellte integrierte Ansatz führt Studierende schrittweise in *Essence*-Konzepte ein, lässt sie diese in einer virtuellen Spielumgebung erproben und schließlich erfolgreich praktisch in SE-Aufgaben einsetzen. Dabei unterstützt eine effiziente und vielseitige Lernumgebung bei der aktiven Konstruktion von Wissen. Sie regt die Eigenaktivitäten der Lernenden an, ermöglicht die Betrachtung des Lerngegenstands aus verschiedenen Perspektiven und fördert früh Artikulation und Reflexion im sozialen Austausch. Ziel dieses Ansatzes ist es, Studierende des SE für die Vielfalt der zu berücksichtigenden Dimensionen einer SE-Aufgabe zu sensibilisieren und ihnen darüber hinaus eine wertvolle Orientierungshilfe für die Verwendung von SE-Methoden innerhalb und außerhalb des Curriculums zu erschließen. Dabei wird großer

Wert auf die Übertragbarkeit der erlangten Erkenntnisse auf andere Kontexte gelegt.

Einleitung

Probleme / Herausforderungen

Ein idealtypischer Grundsatz konstruktivistischer Didaktik lautet: „Jeder Sinn, den ich selbst für mich einsehe, jede Regel, die ich aus Einsicht selbst aufgestellt habe, treibt mich mehr an, überzeugt mich stärker und motiviert mich höher, als von außen gesetzter Sinn, den ich nicht oder kaum durchschaue...“ (Reich, 2008: S. 95).

Die Notwendigkeit von definierten Prozessen und Methoden in der Softwareentwicklung erschließt sich Studierenden erfahrungsgemäß nicht zwingend intuitiv. Häufig besteht hier die Auffassung, dass (überschaubar komplexe) eigene Projekte in der Vergangenheit mehr oder weniger gut auch ohne "einschränkende Vorschriften" zu bewältigen waren. In Vorlesungen wird eine tiefergehende Diskussion, Analyse bzw. der Vergleich von SE-Methoden durch einen meist (noch) relativ engen Erfahrungshorizont erschwert und begrenzt. In Kursprojekten, welche Lehrveranstaltungen begleiten oder abschließen, können häufig folgende Beobachtungen angestellt werden:

- Es fällt Studierenden nicht leicht, sich in einer gegebenen SE-Methode zu orientieren.
- Es fällt Studierenden schwer, die Frage(n) WER, WIE, WANN und insbesondere WARUM bestimmten Aktivitäten nachgehen sollte, zu beantworten.
- Eine an sich sinnvolle und gewollte Arbeitsteilung innerhalb der Projektteams lenkt den Fokus der Beteiligten auf die Er-

stellung von geforderten Artefakten und führt schnell zur Spezialisierung Einzelner im Team. Es entsteht eine Fixierung auf fachliche und/oder technologische Details. Dabei geht mehrheitlich der Blick für "das große Ganze" der SE-Aufgabe verloren.

- Die gewonnen Erkenntnisse lassen sich häufig nur schwer auf andere Kontexte und kommende Herausforderungen übertragen.

Die Gesellschaft für Informatik e.V. empfiehlt, „... nicht nur gegenwartsnahe Inhalte zu vermitteln sondern auch theoretisch untermauerte Konzepte und Methoden, die über aktuelle Trends hinweg Bestand haben. ... Dabei wird ... ‚strukturelles‘ ... Denken verlangt. ... ‚Strukturelles‘ Denken ... ist mehr-dimensional, es erfordert die gleichzeitige Erfassung mehrerer Entitäten mit ihren strukturellen und Verhaltens-Beziehungen.“ (Gesellschaft für Informatik e.V. (GI), 2005: S. 8)

Fragestellungen

Mit *Essence* liegt nun ein Standard (Object Management Group, 2014) vor, welcher verspricht, die essentiellen Dimensionen einer jeden SE-Aufgabe (*endeavour*), die notwendigen Aktivitäten und Kompetenzen in einem kompakten, universellen, erweiterbaren und ausführbaren Kernel zusammenzufassen (Object Management Group, 2014). Daraus erwachsen folgende Fragestellungen:

- **Fragestellung 1:** Sind *Essence* und der *Essence Kernel* geeignet, Studierende theoretisch und praktisch in SE-Methoden einzuführen?
- **Fragestellung 2:** Wie unterstützen *Essence* und der *Essence Kernel* aufgrund ihrer Charakteristik Lehrende in didaktischer und organisatorischer Hinsicht?

Die Förderung eines tiefen Verständnisses von Inhalten und deren Aufnahme in den eigenen Wertekanon sollte Ziel der SE-Ausbildung sein (Ludewig, 2009). Um sich den tiefen Sinn eines Themas zu erschließen, bedarf es einer aktiven kognitiven Auseinandersetzung mit diesem. Daraus folgt:

- **Fragestellung 3:** Wie kann eine Lernumgebung gestaltet werden, welche den Einstieg in *Essence* erleichtert, auf den praktischen Einsatz vorbereitet und ein möglichst tiefes Verständnis fördert?

Gliederung

Im Folgenden untersucht dieser Beitrag die o.g. Fragestellungen. Da die *Essence*-Spezifikation vergleichsweise jung ist, werden vorab zentrale Konzepte und Elemente der Spezifikation und des *SEMAT Essence Kernel* in Kürze vorgestellt. Der Beitrag schließt mit einem Fazit, gibt einen kurzen Ausblick und beschreibt die nächsten geplanten Schritte des Sim4SEEd-Projekts.

SEMAT Essence

Die Spezifikation „*Kernel and Language for Software Engineering Methods (Essence)*“ (Object Management Group, 2014) ist ein Ergebnis der *SEMAT Initiative* (SEMAT, 2014). Sie liegt aktuell in der Version 1.0 beta 2 vor und definiert einen Kernel sowie eine Sprache für die Erzeugung, Nutzung und Verbesserung vorliegender und zukünftiger SE-Methoden. Die Trennung zwischen Sprache und

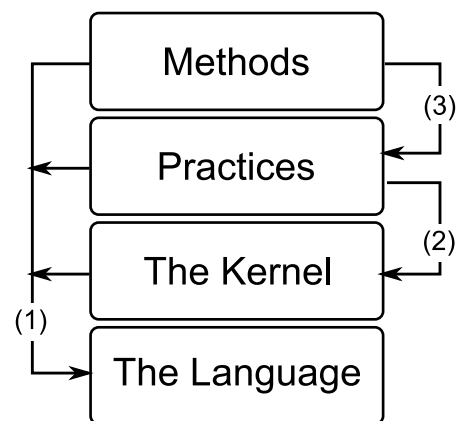


Abb. 1: SEMAT Essence: struktureller Aufbau

Kernel unterscheidet *Essence* von vorangehenden Ansätzen wie *SPEM* (Object Management Group, 2008) und *ISO 24744* (International Organization for Standardisation (ISO), 2007). *Essence* zielt nicht vorrangig auf Prozessingenieure sondern auf SE-Praktiker (Object Management Group, 2014).

Abbildung 1 stellt den strukturellen Aufbau des Standards dar. Die Sprache (*Language*) liefert mit einem Metamodell die syntaktische Infrastruktur, mit welcher die grundlegenden Konzepte in einem Kernel abgebildet werden (1). Auf Basis dieser Konzepte werden Praktiken (*Practices*) definiert (2), welche schließlich zu Methoden (*Methods*) zusammengesetzt werden können (3). Der *Essence Kernel* verfolgt nicht das Ziel, jedes denkbare Detail einer jeden verfügbaren SE-Methode abzubilden. Sein Ziel ist es, so klein und universell wie möglich die Essenz dessen abzubilden, was jeder SE-Aufgabe innewohnt. Dazu verwendet der *Essence Kernel* die Konzepte von *Alphas*, *Activity Spaces* und *Competencies*.

Alphas (Abkürzung für *Abstract-Level Progress Health Attribute*) repräsentieren die essentiellen Elemente, deren Fortschritt (*Progress*) und Gesundheit (*Health*) ein jedes Team andauernd im Blick

und *Endeavour*) ersichtlich, welche für eine Strukturierung sorgen. Jedes *Alpha* verfügt über eine Menge definierter Zustände, sog. *Alpha States*. Ziel einer jeden SE-Aufgabe ist es, für jedes *Alpha* von einem Anfangszustand über dessen Folgezustände hin zu einem gewünschten Endzustand zu gelangen. Dabei müssen die *Alphas* balanciert weiterentwickelt werden. Eine Checkliste mit *Checkpoints* für jeden *Alpha State* dient als Basis für die Feststellung des aktuellen Zustands eines *Alpha*. Ausgehend vom aktuellen Zustand der *Alphas* ergeben sich deren angestrebten Folgezustände. Die *Alphas* des *Essence Kernel* sind sehr universell gehalten und werden häufig nicht für die detaillierte Beschreibung einer SE-Methode ausreichen. Daher kann der *Essence Kernel* durch Kernelerweiterungen flexibel nach Bedarf ergänzt werden. Die in der Spezifikation enthaltene *Development Extension* bspw. fügt die Sub-*Alphas* *Requirement Item*, *Software System Ele-*

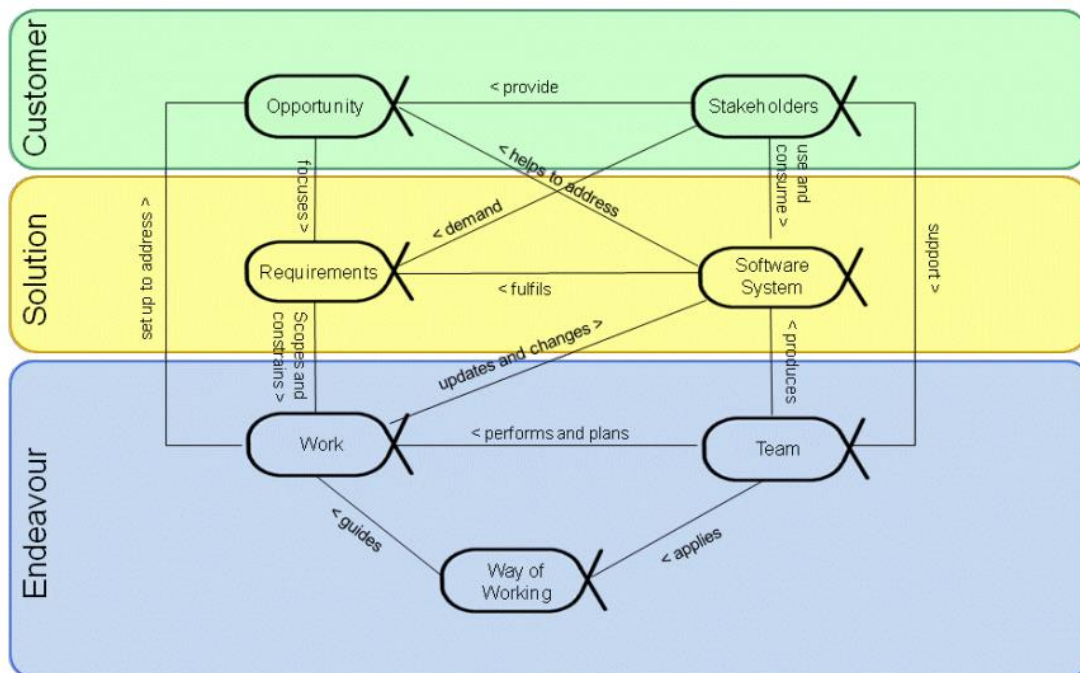


Abb. 2: *Essence Kernel: Alphas* und deren Beziehungen (Object Management Group, 2014)

behalten sollte. Sie beschreiben die Dinge, mit welchen immer gearbeitet wird, sobald Software entwickelt, gewartet oder unterstützt wird. Abbildung 2 stellt die 7 *Alphas* des Kernel sowie deren Beziehungen untereinander dar. Die *Alphas* des *Essence Kernel* sind: *Stakeholder*, *Opportunity*, *Requirements*, *Software System*, *Team*, *Work* und *Way of Working*. Aus der Abbildung ist auch die Organisation des Kernel in 3 sog. *Areas of Concern* (*Customer*, *Solution*

ment und *Bug* ein, welche die Kernel *Alphas* *Requirements* und *Software System* weiter strukturieren und genauer beschreiben. Mit ihrer Verwendung kann zur Laufzeit eines Softwareprojekts nicht mehr nur der Fortschritt aller Anforderungen (*Requirements*) in ihrer Gesamtheit, sondern jeder einzelnen Instanz eines Anforderungselements analysiert und gesteuert werden.

Tabelle 1 stellt in Spalte 3 die *Activity Spaces* des *Essence Kernel* dar. Diese dienen als abstrahierte Platzhalter für konkrete Aktivitäten. Jeder *Activity Space* zielt auf die Erreichung eines oder mehrerer

eines bestimmten *Alpha State*. Diese Beziehungen sind in Abbildung 3 zusammengefasst.

Die freie Komposition von *Practices* erfolgt in Methoden (*Methods*). *Essence* fordert dazu auf, auch

Area of Concern	Alphas	Activity Spaces	Competencies
Customer	<ul style="list-style-type: none"> – Opportunity – Stakeholders 	<ul style="list-style-type: none"> – Explore Possibilities – Understand Stakeholders Needs – Ensure Stakeholder Satisfaction – Use the System 	<ul style="list-style-type: none"> – Stakeholder Representation
Solution	<ul style="list-style-type: none"> – Requirements – Software System 	<ul style="list-style-type: none"> – Understand the Requirements – Shape the System – Implement the System – Test the System – Deploy the System – Operate the System 	<ul style="list-style-type: none"> – Analysis – Development – Testing
Endeavour	<ul style="list-style-type: none"> – Work – Team – Way of Working 	<ul style="list-style-type: none"> – Prepare to do the Work – Coordinate Activity – Support the Team – Track Progress – Stop the Work 	<ul style="list-style-type: none"> – Leadership – Management

Tabelle 1: *Essence Kernel*: Die *Areas of Concern* und ihre zugeordneten *Alphas*, *Activity Spaces* und *Competencies*

Alpha States. Aktivitäten erfordern spezifische Kompetenzen (*Competencies*) auf einem definierten Kompetenzniveau (*Competency Level*). Der *Essence Kernel* definiert 6 Kompetenzen mit verschiedenen Niveaus. Diese sind ebenfalls in Tabelle 1 dargestellt.

Da der Kernel unabhängig von jeder SE-Methode definiert ist, fehlen hier designbedingt methodenspezifische konkrete Aktivitäten (*Activities*) und Arbeitsergebnisse (*Work Products*). Diese werden erst in Praktiken (*Practices*) auf Basis des Kernels definiert. *Practices* stellen wiederholbare Ansätze dar, begrenzte Aspekte innerhalb einer SE-Aufgabe systematisch und verifizierbar mit einem spezifischen Ziel anzugehen. Eine *Practice* könnte bspw. die Verwendung von *User Stories* sein, um Benutzeranforderungen festzuhalten. Die in *Practices* definierten konkreten Aktivitäten sind jeweils einem *Activity Space* des Kernels zugeordnet. *Activity Spaces* dienen somit der Organisation von *Activities*. Arbeitsergebnisse (*Work Products*) können auf verschiedenen Detaillierungsstufen (*Levels of Detail*) definiert werden. Sie beschreiben praxisspezifisch *Alphas* und dienen als Nachweis für die Erreichung

während einer laufenden SE-Aufgabe agil auf neue Entwicklungen und Erkenntnisse zu reagieren, und *Practices* bei Bedarf in die verwendete SE-Methode einzuführen oder auszutauschen.

Die *Essence-Spezifikation* in ihrer Breite und Tiefe vorzustellen, würde den Rahmen dieses Beitrags sprengen. Hier sei auf weitere Literatur verwiesen (Jacobson u. a., 2013; Object Management Group, 2014; Striwe, Goedicke, 2013).

Kritik

Es soll nicht unerwähnt bleiben, dass die SEMAT-Ideen nicht ohne Kritiker waren (Alistair Cockburn, 2007; Fowler, 2010; Ivar Jacobson u. a., 2010). Ein Großteil der Kritik beruht auf frühen Missverständnissen über die Ausrichtung der SEMAT Initiative. Ein relativ häufiges Missverständnis beruht darauf, *Essence* mit einer bestimmten SE-Methode, z.B. Scrum oder RUP, zu vergleichen. "But the comparisons [...] should not be Scrum versus Essence, or CMMI versus Essence, or RUP versus Essence. But rather they should be Scrum versus (Scrum + Essence), or CMMI versus (CMMI + Essence), or RUP versus (RUP + Essence). Essence is not in competition with any existing practice or

method. It is agnostic to your chosen approach and can be implemented by your team today without changing what you are currently doing. It is more about helping you do what you are already doing so you can keep doing it even better in the future." (Paul E. McMahon, 2013)

Sind *Essence* und der *Essence Kernel* geeignet, Studierende theoretisch und praktisch in SE-Methoden einzuführen?

Bei der Beantwortung dieser Frage sollen zwei Aspekte besondere Berücksichtigung finden:

- Wie erleichtert *Essence* den Zugang zu SE-Methoden?
- Welche zusätzlichen Kompetenzen können mit *Essence* vermittelt werden?

Der folgende Abschnitt widmet sich der Beantwortung dieser beiden Fragen.

Wie erleichtert *Essence* den Zugang zu SE-Methoden?

Die Verbreitung von SE-Methoden in unterschiedlichsten Formaten verwehrt Einsteigern einen einheitlichen Zugang. "This can get very confusing [...] as the lack of any common ground between the sources can lead to people saying the same thing in different ways and different things in the same way." (Jacobson u. a., 2013: S. 55) *Essence* bietet ein gemeinsames Vokabular und mit dem Kernel einen einheitlichen Bezugspunkt.

Die Beschreibung von Praktiken auf Basis des Kernels, erleichtert deren Einordnung. Ist der Kernel bekannt, so bezieht sich jede *Essence*-Praktik auf etwas bereits Bekanntes, was das Erlernen einer Praktik nachhaltig erleichtert.

"[*Essence* promotes] learning and training focused on the essentials..." (Jacobson u. a., 2013: S. 48) - mit *Essence* und dem *Essence Kernel* liegt der Fokus zuerst auf dem Universellen und Essentiellen. Auch ohne alle Details der *Essence Language* oder einer speziellen SE-Methode zu kennen, lassen sich die Konzepte des kompakten *Essence Kernel* sehr gut in jeder SE-Aufgabe verwenden. Die Einstiegshürde – auch zur teilweisen Verwendung – liegt damit sehr viel tiefer. *Essence* priorisiert die tägliche Anwendbarkeit vor einer allumfänglichen detaillierten Prozessbeschreibung und wendet sich damit an eine viel breitere Zielgruppe als vorange-

gangene Ansätze (International Organization for Standardisation (ISO), 2007; Object Management Group, 2008).

Der Kernel ist auch im physischen Sinne „greifbar“ (*tangible*). Über eine Kartenmetapher lassen sich die wichtigsten Aspekte der *Alphas* und deren *Alpha States* als Karten – ganz wie in einem Kartenspiel – betrachten und tatsächlich anfassen. *Essence* definiert hierzu die Anatomie verschiedener Kartentypen. Drückt man sich die Karten des Kernel oder einer SE-Methode aus, lässt sich bei Diskussionen sehr agil damit arbeiten. So werden bspw. *Alpha State*-Karten einfach verschoben, um gegenwärtige und angestrebte Situationen zu visualisieren. Dies verdeutlicht, dass es sich nicht „nur“ um abstrakte Konzepte, sondern um handfeste praktische Unterstützung bei der Bewältigung einer SE-Aufgabe handelt.

„[*Essence*] focuses on the needs of the software professional and values the ‘use of methods’ over ‘the description of method definitions’ (the normal priority in the past).“ (Jacobson u. a., 2012: S. 10) Der *Essence Kernel* ist ausführbar (*actionable*). Er beschreibt nicht nur, was getan werden sollte, sondern dokumentiert über die erreichten *Alpha States* auch was tatsächlich bereits getan wurde. Die erlernten Konzepte lassen sich praktisch, bspw. in einem Kursprojekt, erleben. Dabei lässt sich die Frage „Wo stehen wir gerade?“ über die aktuellen *Alpha States* jederzeit schnell beantworten. Das unterstützt Studierende, welche sich i.d.R. nicht exklusiv um ein Kursprojekt kümmern, sondern parallel andere Lehrveranstaltungen besuchen.

Das Konzept der *Alphas* und ihrer Zustände (*Alpha States*) fordert die andauernde Auseinandersetzung mit den essentiellen Dimensionen der SE-Aufgabe. Fortschritt und Gesundheit aller relevanten Dimensionen werden strukturiert und regelmäßig bewertet. In Kursprojekten bietet dies einen hervorragenden Anlass, im Team über den aktuellen Stand zu diskutieren und neue Ziele zu vereinbaren – ohne sich dabei ausschließlich in fachlichen oder technischen Details des Projekts zu verlieren.

Der *Essence Kernel* hilft dabei, Diskussionen im Team zu steuern. Die verfügbaren Checklisten halten dazu an, über Bereiche nachzudenken, welche zu Problemen heranwachsen könnten – auch dann, wenn das Team dies aus eigener Erfahrung noch nicht vorhersehen würde. Die Checklisten helfen, in der Diskussion die richtigen Fragen zu stellen, um zu besseren Entscheidungen zu gelangen.

Die Beziehungen zwischen *Alphas*, *Alpha States*, *Activity Spaces*, *Activities*, *Work Products* und *Competencies* ergänzt um das Konzept der *Patterns* (sichtbar in Abbildung 3) sorgen dafür, dass sich ausgehend von den aktuellen Zuständen der *Alphas* die notwendigen Aktivitäten zur Erreichung der als nächstes angestrebten Zustände ableiten lassen.

Es ist die Aufgabe von SE-Methoden, zu beschreiben, WER WAS WANN in einer SE-Aufgabe tun sollte. Im Hinblick auf ein angestrebtes tiefes Verständnis, ist insbesondere die Frage nach dem WARUM von Bedeutung. *Essence* bietet hier mit dem Bezug der *Activities* und *Work Products* auf *Alpha States* ein methodenübergreifendes einheitli-

chen für jede Aktivität verdeutlichen, dass es auch außerhalb der *Solution Area of Concern* wichtige Betätigungsfelder gibt.

Welche zusätzlichen Kompetenzen können mit *Essence* vermittelt werden?

Im Berufsleben werden Studierende auf eine Vielzahl von SE-Methoden und Praktiken treffen. Die Kompetenz, diese neuen Praktiken und Methoden auf einer gemeinsamen Basis einzuordnen, Gemeinsamkeiten und Unterschiede zu erkennen, um daraus eigene Schlüsse zu ziehen, wird im Berufsleben von großem Wert sein. Sie erleichtert die Kommunikation zwischen verschiedenen Projektteams und Organisationen, welche unterschiedliche

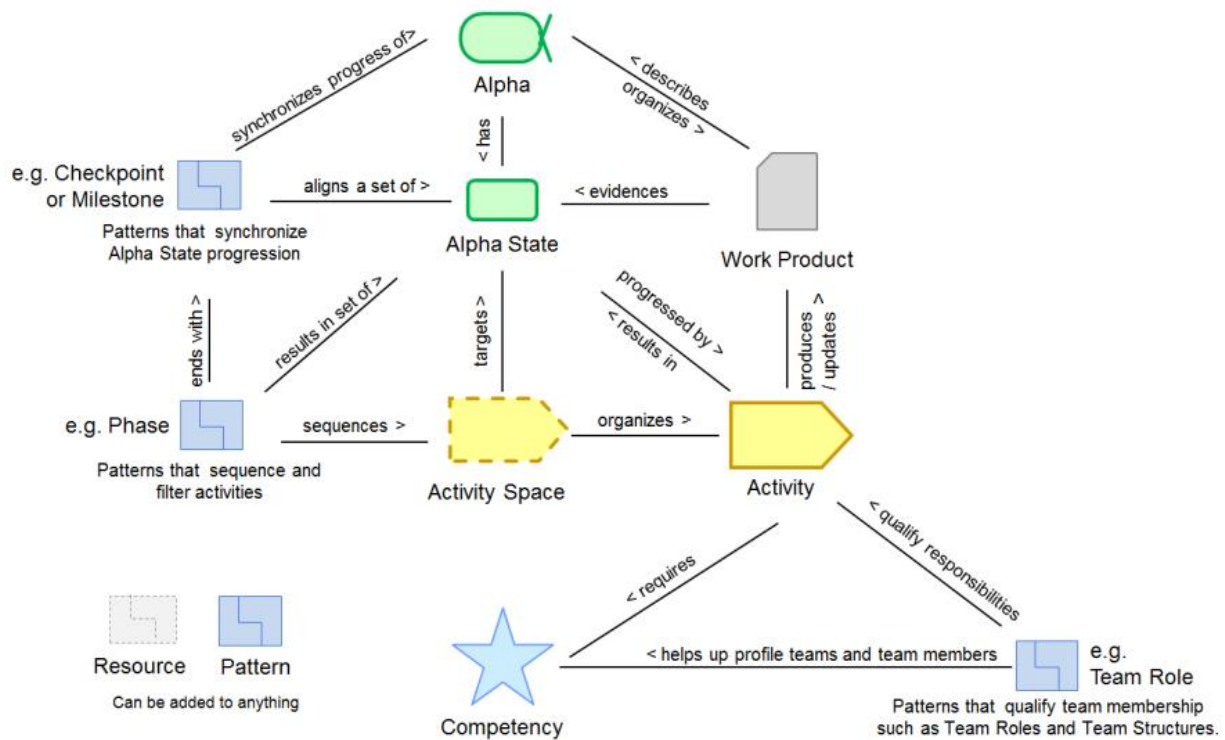


Abb. 3: Essence Language: konzeptioneller Überblick (Object Management Group, 2014)

ches Konzept.

Damit liefert *Essence* neben einem einheitlichen Vokabular in vielfacher Hinsicht Orientierung und strukturiertes direkt einsetzbares Expertenwissen in Form von Checklisten. Die Greifbarkeit über die Kartenmetapher nimmt dem Lerngegenstand etwas von seiner Abstraktheit. Der Kernel dient als Wegweiser, regt fortlaufend zur Reflexion an, WARUM etwas getan wird und fordert die Betrachtung aller essentiellen Dimensionen. Die definierten Kompe-

tenzen für jede Aktivität verdeutlichen, dass es auch außerhalb der *Solution Area of Concern* wichtige Betätigungsfelder gibt.

Die fortwährende Berücksichtigung aller relevanten Dimensionen in Form von *Alphas* und ihren *Alpha States* schärft die Kompetenz, über die aktuell wichtige Detailaufgabe hinauszuschauen und „das große Ganze“ im Blick zu behalten. Checklisten, welche helfen, die richtigen Fragen zu stellen, unterstützen beim Aufbau einer kritischen Denkweise. Es wird zu zielgerichteter, strukturierter und

verifizierbarer Arbeitsweise angehalten – mit einem starken Fokus auf dem Zweck jeder Aktivität.

Die freie Kombination von *Best Practices* zu *Methods* schafft mehr Flexibilität und Freiheit als eher monolithisch anmutende vorangehende Ansätze. Die Einbindung neuer Ideen und Praktiken in bestehende SE-Methoden und in laufenden Softwareprojekten wird dadurch erleichtert. Vermittelt wird hier die Kompetenz, die erlernten Konzepte auf andere Kontexte zu übertragen.

Wie unterstützen *Essence* und der *Essence Kernel* aufgrund ihrer Charakteristik Lehrende in didaktischer und organisatorischer Hinsicht?

Verschiedene SE-Methoden und Softwareprozesse sind unterschiedlich gut für bestimmte Anwendungskontexte geeignet. Ohne sich früh auf einzelne Anwendungskontexte festzulegen, können auf Basis des *Essence Kernel* verschiedene Ansätze vorgestellt werden. *Essence* gestattet es dabei sehr gut, den Detaillierungsgrad einer SE-Methode zu gestalten. So können Anzahl, Breite und Tiefe der vorgestellten SE-Methoden flexibel an den Bedarf von Lehrveranstaltungen angepasst werden.

Veranstaltungsübergreifend bietet *Essence* mit dem Kernel eine gemeinsame Basis und Anknüpfungspunkte. Sind die Grundkonzepte erlernt, so lassen sich verschiedene Praktiken und Methoden didaktisch daraus entwickeln.

Die Arbeit mit *Essence* wird bereits durch digitale Werkzeuge unterstützt, ist jedoch nicht an diese gebunden. So wird eine Vielzahl von Lernarrangements ermöglicht, welche den Lerngegenstand weniger abstrakt erscheinen lassen.

Neu auftauchende SE-Praktiken werden sich auf Basis des Kernels leichter in SE-Methoden integrieren und vermitteln lassen.

Wie kann eine Lernumgebung gestaltet werden, welche den Einstieg in *Essence* erleichtert und auf den praktischen Einsatz vorbereitet?

In der Antwort auf eine Kritik bemerkte einer der SEMAT Initiatoren, „... be prepared to read carefully since it unfortunately takes time to get under the skin of the ideas.“ (Alistair Cockburn, 2007) Wie lässt sich im Rahmen der SE-Ausbildung der Zugang und Einstieg in *Essence* und den *Essence*

Kernel erleichtern? Der folgende Abschnitt stellt einen integrierten Ansatz vor, welcher auf die Bereitstellung einer geeigneten Lernumgebung abzielt, um die Eigenaktivitäten der Lernenden anzuregen und die aktive Konstruktion von Wissen zu unterstützen. Die Betrachtung des Lerngegenstands aus verschiedenen Perspektiven, Artikulation und Reflexion im sozialen Austausch tragen grundlegend zum erfolgreichen Lernen bei (Kritzenberger, 2005). Daher stellen diese Aspekte zentrale Ziele des entworfenen Ansatzes dar.

Warum „spielerisch“ und „integriert“?

„Integriert“ meint im Sinne der Duden-Definition „so beschaffen, dass Unterschiedliches, Verschiedenartiges miteinander verbunden, vereinigt ist“. Im Ansatz, welcher in diesem Beitrag vorgestellt wird, werden aktivierende spielbasierte Konzepte und Simulation dort eingebunden, wo sie aufgrund ihrer Charakteristik einen Beitrag zur Erreichung der Lernziele leisten. Der Begriff „integriert“ passt an dieser Stelle nur bedingt, denn wie der folgende Abschnitt zeigt, sind Spielen und Lernen grundsätzlich nichts Verschiedenartiges. Simulation und spielbasierte Ansätze unterstützen in den Phasen, wo es ohne sie traditionell an Interaktivität mangelt, wo die aktive Konstruktion von Wissen sonst nicht optimal unterstützt wird. Sie tragen zur Erreichung der Lernziele bei, indem Sie eine motivierende, fesselnde Lernumgebung schaffen, die zur aktiven Auseinandersetzung mit einem traditionell eher abstrakten Lerngegenstand einlädt.

In einem Kursprojekt, in welchem die Lernenden aktiv sind, bedarf es aus dieser Sicht keiner weiteren Aktivierung. Hier liefert der vorgestellte Ansatz jedoch Orientierungshilfe und Anlässe zur Artikulation, Diskussion und Reflexion, welche erneut der aktiven Wissenskonstruktion dienen.

Die Phase der theoretischen Einführung und die Phase des praktischen Einsatzes von SE-Methoden werden in diesem Ansatz durch eine Phase verbunden, in welcher die kennengelernten Konzepte in einem Simulationsspiel virtuell erprobt und vertieft werden. Werkzeuge aus dieser Phase des virtuellen Einsatzes finden auch im praktischen Einsatz Verwendung, so dass hier ein weiteres verbindendes Element besteht.

Damit integriert dieser Ansatz Theorie- und Praxisphasen und bindet unterstützende Elemente aus spielbasierten Konzepten ein. Er gibt den Studierenden die Möglichkeit, schrittweise auf Be-

kanntes aufzubauen, das erlangte Verständnis zu erweitern, es zu vertiefen und auf andere Kontexte zu transferieren.

Simulation und digitale Spiele in der Software Engineering Ausbildung

In einem Beitrag der SEUH 2013 (Jöran Pieper, 2013) wurden Vorzüge und Beispiele für den Einsatz von Simulation und Digital Game-Based Learning (DGBL) für den Bereich der Softwareprozesse in der SE-Ausbildung beschrieben. Zusammenfassend lässt sich festhalten, dass Spiel und Nachahmung als natürliche Lernstrategien lebenslang wichtige Akkommodations- und Assimilationsstrategien bleiben (Rieber, 1996). Der fesselnde und motivierende Charakter von Spielen fördert die intrinsische Motivation der Lernenden, indem er sie dazu motiviert, die Verantwortung für den eigenen Lernfortschritt zu übernehmen (Akilli, 2007). Jede Verbindung von Ausbildungsinhalt (*educational content*) und digitalen Spielen wird dabei als *Digital Game-Based Learning (DGBL)* bezeichnet (Prensky, 2007). DGBL fördert den Lernprozess, indem durch die Integration attraktiver Spielelemente – wie Interaktivität, Herausforderungen, kontinuierlichem Feedback, Multimedialität, dem Gefühl der Selbstwirksamkeit, Wettbewerb und Belohnungen – eine motivierende und fesselnde Lernumgebung bereitgestellt wird. „Digitale Spiele können demnach [...] ein selbstgeleitetes Lernen durch Exploration ermöglichen und befördern.“ (Breuer, 2010: S. 12)

Bisherige Forschungsergebnisse zeigten, dass die sorgfältige Planung und Einbettung von Spielaktivitäten in das Curriculum, ausreichende Anleitung, detailliertes Feedback, Diskussion sowie die gründliche Auswertung und Erklärung von Spielresultaten eine essentiell wichtige Rolle für einen Lernerfolg spielen (Wangenheim, Shull, 2009).

Integrierter Ansatz

Der hier vorgestellte integrierte Ansatz gliedert sich in 3 + X Phasen. Die einzelnen Phasen bauen aufeinander auf. Die Phasen 1 bis 3 dienen dabei der Einführung in *Essence* und den *Essence Kernel* sowie der praktischen Erprobung in virtuellen und realen Projektumgebungen. Die Phasen 4 bis X dienen der weiteren Vertiefung und lassen die Lernenden über die Rolle von *Essence Method*-Konsumenten hinauswachsen.

Phase 1: Den *Essence Kernel* kennenlernen

In dieser Phase machen sich Studierende mit *Essence* und dem *Essence Kernel* vertraut. Ziele, Vorteile und die grundsätzliche Struktur des Kernels werden kennengelernt. Als passende Lernmethoden dienen hierbei Literaturstudium, Fallstudien sowie Lehrvorträge in der klassischen Vorlesung. In dieser Phase soll das Interesse geweckt und zur aktiven Auseinandersetzung mit dem Thema angeregt werden. Zum Literaturstudium sind zahlreiche Quellen, einschließlich der gut lesbaren Spezifikation, verfügbar. Auch Fallstudien zur Verwendung des *Essence Kernel* in verschiedenen Projektphasen sind zu finden (Jacobson u. a., 2013).

Für ein interaktives Kennenlernen des *Essence Kernel*, insbesondere der *Alphas* und ihrer Beziehungen untereinander, wurde im Sim4SEEd-Projekt ein Lernspiel entwickelt, welches sich ohne weitere Installationen im Browser ausführen lässt. Spieler sollen mit Hilfe des Spiels in die Lage versetzt werden, die Elemente des *Essence Kernel* zu benennen, Definitionen korrekt zuzuordnen und Beziehungen zwischen den Elementen herzustellen.

Um die Vorlesung in dieser Phase interaktiv, motivierend und fesselnd zu gestalten, eignet sich das kostenfrei online verfügbare *Kahoot!* (Kahoot! AS, 2014) hervorragend. *Kahoot!* ist ein „*game-based blended-learning and classroom response system*“ und integriert dabei spielbasierte und soziale Ansätze. Mit *Kahoot!* lassen sich Quiz, Umfragen und Diskussionen vorbereiten und live in die Präsenzveranstaltung integrieren. Es bedarf dafür keiner besonderen Infrastruktur. Der bei einem Quiz entstehende Wettbewerbscharakter weckt Emotionen und wirkt nach Erfahrungen des Autors überaus motivierend. Studierende erhalten unmittelbar Feedback über ihren individuellen Lernstand. Lehrende erhalten unmittelbar einen Eindruck, wie gut einzelne Sachverhalte in der Kursgruppe bereits verstanden wurden. Falsch beantwortete Fragen bieten Anlässe zur Diskussion und können dazu beitragen, Missverständnisse oder Wissenslücken aufzudecken und zu beseitigen. Um den Einstieg zu erleichtern, stellt das Sim4SEEd-Projekt bereits ein vorbereitetes frei verfügbares *Kahoot!*-Quiz zum *Essence Kernel* bereit.

Zum Ende dieser Phase sollten allen Teilnehmenden die grundlegenden Konzepte und Elemente des *Essence Kernel* bekannt sein. Je nach Fokus wurden evtl. schon einzelne Konzepte vertieft.

Phase 2: *Essence* virtuell praktizieren

In dieser nächsten Phase werden die erlernten Konzepte angewendet, um deren Zusammenspiel zu verdeutlichen und die bereits erlangten Erkenntnisse zu vertiefen. Als Lernmethode wird hier ein interaktives kollaboratives Simulationsspiel genutzt, welches im Rahmen des Sim4SEEd-Projekts entwickelt wird.

Lernziel dieses Simulationsspiels ist es, sich in einem simulierten Softwareprojekt mit Hilfe des *Essence Kernel* und einer zuvor gewählten SE-Methode, welche auf Basis des Kerns definiert wurde, zurechtzufinden. Dabei stoßen die Spieler auf Kernel-Erweiterungen und Praktiken (*Practices*), welche den *Essence Kernel* methodenspezifisch ergänzen. Spieler sollen dabei die vorgegebene SE-Methode – wie in einem Adventure-Spiel – erkunden und die entdeckten Methodenelemente einsetzen. Dabei ist es wichtig, die richtigen Dinge (*Activities*) in der richtigen Reihenfolge zu tun und über alle relevanten Dimensionen (*Alphas*, *Alpha States*) der Aufgabe nachzudenken. Die Spieler erkennen dabei, dass unterschiedliche Aktivitäten (*Activities*) verschiedene Kompetenzen erfordern.

Die Einschätzung der *Alpha States* wird durch die Rückmeldung der virtuellen Spielfiguren ermöglicht. Dabei orientieren sich deren Rückmeldungen an den *Checkpoints* der *Alpha State*-Checklisten aus der gewählten SE-Praktik.

Durch die Zuordnung von Aufgaben an das virtuelle Projektteam, die aktive Verfolgung der Rückmeldungen sowie Analyse der *Alpha State*-Checklisten kann ein Spieler den Verlauf des virtuellen Projekts steuern. Spielfiguren stellen im Spiel wiederkehrend Fragen zu zugeteilten Aktivitäten, *Alphas* etc. Die richtige Beantwortung der Multiple Choice Fragen wirkt sich dabei positiv auf das Spielergebnis aus.

Um sich in der SE-Methode zu orientieren nutzen die Spieler den *Essence Method & Alpha State Navigator*, welcher im Sim4SEEd-Projekt entwickelt wird. Über diesen ist es möglich, durch die SE-Methode zu browsen und die Dokumentation der einzelnen Methodenelemente abzurufen. Gleichzeitig bietet er die Möglichkeit, die *Alphas* im Verlauf der SE-Aufgabe auf die jeweils eingeschätzten *Alpha States* zu setzen. Dabei unterstützen die hinterlegten Checklisten eines jeden *Alpha State*.

Der *Essence Method & Alpha State Navigator* wird mit dem Ziel entwickelt, sowohl in der simulierten

Spielwelt, als auch in einem echten Projekt eingesetzt werden zu können. Er dient damit als weiteres verbindendes Element zwischen dieser und der nächsten Phase 3.

Die Konstruktion des dem Spiel zugrunde liegenden Simulationsmodells erfolgt transparent in mehreren Stufen. In der ersten Stufe wird eine verfügbare *Essence Method* ausgewählt. Alternativ wird auf Basis des *Essence Kernel* mit Kernelerweiterungen, verfügbaren oder selbst kreierten *Practices* werkzeugunterstützt selbst eine *Essence Method* definiert. Im Projekt wird dafür gegenwärtig die *EssWork Practice Workbench* (Ivar Jacobson International, 2014) verwendet. Die Verwendung eines Standards und eines Standardwerkzeugs wird im Projekt gegenüber anderen Ansätzen klar favorisiert, denn ein Lernaufwand an dieser Stelle nützt weit über diese Simulations- und Spielumgebung hinaus.

In der nächsten Stufe wird aus der gewählten SE-Methode ein Simulationsmodell generiert. Das Simulationsmodell folgt dabei der Philosophie des *Essence Kernel*. Es ist ebenso konkret und detailliert, wie die gewählte Methode. Enthält die gewählte Methode bspw. detaillierte *Activities* zur Erreichung der *Alpha States*, so werden diese auch im Simulationsmodell abgebildet. Enthält die gewählte Methode keine *Activities* für einen *Activity Space*, so wird stattdessen der generischere *Activity Space* verwendet. Je nach Fokus und Lernziel kann so gezielt die Komplexität des Modells und damit die Komplexität des Spiels gesteuert werden. Anschließend erfolgen notwendige Quantifizierungen innerhalb des Simulationsmodells. Hier wird bspw. festgelegt, wieviel Aufwand in Personenstunden mit einem bestimmten *Competency Level* in eine Aktivität investiert werden muss, um einen bestimmten *Alpha State* oder ein bestimmtes *Level of Detail* eines *Work Product* zu erreichen. Um aus dem Simulationsmodell ein Spiel zu generieren, bedarf es noch eines Spielszenarios, welches bspw. festlegt, in welchem Umfeld das virtuelle Projekt stattfindet und welches virtuelle Personal dafür zur Verfügung steht.

Bisherige Spielansätze im Bereich der Software Engineering Ausbildung setzten auf reine Einzelspielerumgebungen, welche kaum Unterstützung für spielübergreifende Auswertungen und Diskussion bieten. Das Hinzufügen eines Team-Konzepts zum individuellen Spiel jedes Einzelnen schafft hier Mehrwert. Dabei erkunden und bearbeiten alle

Spieler individuell eine ganze SE-Methode. Gleichzeitig jedoch agiert jeder Spieler auch in einem Team und sammelt für dieses Punkte. Indem die Performance des Teams – die Kumulation der Einzelspielerergebnisse – als primärer Erfolgsfaktor herangezogen wird, werden Zusammenarbeit und Diskussion innerhalb der Teams gefördert. Dashboards, Einzel- und Teamrankings bieten dabei zusätzliche Orientierung. Sie zeigen an, wie gut bisher getroffene Entscheidungen in Relation zu anderen Spielern waren. Diese zusätzliche Orientierung dient als Ausgangspunkt für Interaktionen innerhalb der Teams und als Anlass für den ständigen Versuch, Entscheidungen zu optimieren. Das Simulationsspiel bietet an dieser Stelle eine Qualität, welche in Kursprojekten in dieser komprimierten Form nicht geboten werden kann: Lernende erhalten kontinuierlich Feedback zu einem viel früheren Zeitpunkt, als dies in einem realen (Kurs-)Projekt der Fall wäre. Durch die vorgeschlagene Kombination aus Einzelspiel und Zusammenarbeit im Team, kann parallel auch aus Erfahrungen anderer Spieler gelernt und profitiert werden. Das Spiel bietet damit die Basis für einen „*probe, hypothesize, reprobe, rethink cycle*“ (Gee, 2007: S. 87 ff.), der zu einer tieferen Auseinandersetzung mit der simulierten SE-Methode führt.

Zum Ende dieser Phase haben die Teilnehmenden die erlernten Konzepte bereits einmal selbst in einem definierten simulierten Kontext angewendet. Dafür haben sie sich notwendigerweise in einer SE-Methode orientiert. Die Spielenden haben alle essentiellen Dimensionen (*Alphas*) der SE-Aufgabe betrachtet und andauernd deren Fortschritt und Gesundheit beurteilt. Die *Alphas* und deren *Alpha States* halfen bei der Orientierung und gaben Empfehlungen für die jeweils nächsten Schritte. Das Spielumfeld, welches Zusammenarbeit und Wettbewerb förderte, hat dazu beigetragen, dass Spielansätze und -züge im Team artikuliert und idealerweise reflektiert wurden. Spielergebnisse wurden kursweit ausgewertet und diskutiert. Damit wurden ideale Bedingungen für die nächste Phase geschaffen.

Phase 3: *Essence* in einem echtem Projekt praktizieren

Inhalt der nächsten Phase ist es, *Essence* in einem realen Projekt einzusetzen. Studierende erfahren in einem Kursprojekt einen bemerkenswerten kognitiven Load. Obwohl es sich i.d.R. „nur“ um dem

Kursumfang angepasste Aufgabenstellungen handelt, kommen in einem solchen Projekt viele Dinge zusammen. Dieser kognitive Load kann in Teilen überwältigend wirken. Es besteht die Gefahr, dass in einen Deadline-getriebenen Arbeitsmodus verfallen wird, der allein darauf fixiert ist, die geforderten Artefakte zum jeweiligen Termin zu liefern, ohne über die Arbeitsweise, die ausgeführten Aktivitäten und deren Motivation zu reflektieren.

Mit dem *Essence Kernel*, welcher in den zwei vorangegangenen Phasen kennengelernt wurde, steht dem Projektteam nun ein Werkzeug zur Verfügung, welches Orientierung bietet, was dabei unterstützt, im Entwicklungsprozess allen wichtigen Dimensionen die notwendige Aufmerksamkeit zukommen zu lassen und die richtigen Fragen zu stellen. Wird die gleiche SE-Methode wie in der Phase 2 verwendet, können gleiche Aktivitäten wiedererkannt und noch einmal aus einer anderen Perspektive erlebt werden. Dabei kann der bereits in der vorangegangenen Phase kennengelernte *Essence Method & Alpha State Navigator* eine bekannte Umgebung und Orientierung bieten.

Assessment Poker fördert eine reflektierte Arbeitsweise aller Teammitglieder. Dabei wird von allen Teammitgliedern unabhängig voneinander eine Einschätzung des gegenwärtigen Projektzustands vorgenommen. Jedes Teammitglied denkt dabei über alle relevanten Dimensionen (*Alphas*) nach und schätzt deren aktuellen Zustand (*Alpha State*) ein. Der Vergleich der individuellen Einschätzungen offenbart schnell unterschiedliche Sichtweisen und ist wertvoller Anlass zum Gedankenaustausch. Wird der Projektfortschritt auch von den Lehrenden zu verschiedenen Zeitpunkten (bspw. Meilensteinen) bewertet, so erfolgt hier ein zusätzlicher Abgleich und Feedback. Die Homogenität der individuellen Einschätzungen kann ein Indiz für die mehr oder weniger erfolgreiche Zusammenarbeit im Team sein und erneut als Anlass für reflektierende Diskussionen dienen.

Der *Essence Kernel* wurde international bereits in Lehrveranstaltungen eingesetzt. Es wurde eingeschätzt, dass sich sein Einsatz positiv auf den Lernerfolg ausgewirkt hat: „By matching the project results against the kernel alphas, the students could easily identify the good and bad sides of their development methods. [...] By following all the kernel alphas, the students could learn the total scope of the software-engineering endeavor and

thereby see what would be required of them in their future as professionals.“ (Jacobson u. a., 2012)

Zum Ende dieser dritten Phase haben die Studierenden *Essence* in einem realen Projektumfeld eingesetzt. Anders als im virtuellen Umfeld der Phase 2 wurden hier die in der SE-Methode enthaltenen Praktiken tatsächlich angewendet und ausgeführt. Dabei galt es, auch soziale und teamdynamische Aspekte zu bewältigen. Die beschriebenen Maßnahmen sorgten dafür, dass alle Teammitglieder regelmäßig über den aktuellen Projektstatus nachgedacht haben und regelmäßig im Team über den Fortschritt und die Gesundheit der relevanten Dimensionen reflektiert wurde. Somit kann davon ausgegangen werden, dass an dieser Stelle alle Teilnehmenden grundsätzlich in der Lage sind, eine SE-Methode auf Basis des *Essence Kernel* praktisch anzuwenden.

Phase 4 bis X: *Essence* weiter vertiefen

Je nach Ausrichtung des Curriculums bieten sich weitere Aktivitäten im *Essence*-Umfeld an. Eine naheliegende Vertiefung wäre, *Essence* erneut mit einer anderen SE-Methode virtuell zu praktizieren. Im Anschluss können die Ergebnisse und Erlebnisse der Teilnehmer mit denen aus der Phase 2 und Phase 3 verglichen werden.

Studierende werkzeugunterstützt eine eigene SE-Methode kreieren zu lassen, kann sie aus ihrer Konsumenten- und Anwenderrolle in eine Produzenten-Rolle schlüpfen lassen. Die Erzeugung eigener *Practices* und/oder die Komposition von verfügbaren *Practices* zu einer *Method* sind geeignet, das Verständnis von *Essence* über die reine Anwendung vorgefertigter SE-Methoden und Praktiken hinaus zu vertiefen. Die in der Simulation und im Kursprojekt gewonnenen Erkenntnisse lassen sich dabei verwenden. In der anschließenden Diskussion der erzeugten Methoden können verschiedene Standpunkte und Denkweisen thematisiert werden.

Fazit und Ausblick

Der *SEMAT Essence Kernel* erscheint aufgrund der beschriebenen Charakteristik bestens geeignet, Studierende in die Welt der SE-Methoden einzuführen. Besser als vorhergehende Standards ermöglicht *Essence* einen Einstieg, der sich flexibel an den Anforderungen des jeweiligen Curriculums anpassen lässt. Der Fokus auf die essentiellen Dimensionen von SE-Aufgaben und die Priorisierung der

täglichen Anwendbarkeit für SE-Praktiker als Entwurfsziel der Spezifikation machen *Essence* zu einem wertvollen Werkzeug und Denk-Framework für alle SE-Studierenden. Der universelle Charakter des *Essence*-Standards und des *Essence Kernel* sorgen für eine hohe Übertragbarkeit der erworbenen Kenntnisse auf andere Kontexte innerhalb und außerhalb des SE-Curriculums.

Der hier vorgestellte integrierte Ansatz führt schrittweise in die Konzepte und die Verwendung von *Essence* ein. Er versetzt Studierende in die Lage, schrittweise die enthaltenen Konzepte kennenzulernen, dabei auf Bekanntes aufzubauen und SE-Methoden erfolgreich praktisch in SE-Aufgaben einzusetzen. Um die aktive Konstruktion von Wissen zu unterstützen, wird dabei eine effiziente und vielseitige Lernumgebung bereitgestellt, welche die Eigenaktivitäten der Lernenden anregt, die Betrachtung des Lerngegenstands aus verschiedenen Perspektiven ermöglicht und Artikulation und Reflexion im sozialen Austausch fördert. Simulation und Digital Game-Based Learning (DGBL) sind dabei wichtige Bausteine.

Mit *Essence* und dem *Essence Kernel* kann über die Grenzen einzelner Lehrveranstaltungen hinaus eine gemeinsame Basis verwendet werden. Interessant wäre es zu untersuchen, ob sich auch ein ganzes Informatik-Curriculum mit Hilfe des *Essence Kernel* strukturieren ließe. Auch Nicht-SE-Kernfächer zur Stärkung von Soft Skills, wirtschaftlichen und organisatorischen Kompetenzen etc. sollten sich bspw. über die *Areas of Concern Customer* sowie *Endeavour* abbilden lassen.

Nächste Schritte

Sim4SEEd (www.sim4seed.org) ist ein laufendes Forschungsprojekt. Der derzeitige Fokus der Projektaktivitäten liegt neben der Implementierung eines *Essence*-Simulationsmodells und der Komplettierung eines Simulationsspiels auf dem Einsatz und der Evaluierung des hier vorgestellten Ansatzes. Dazu ist der Einsatz dieses Konzepts mit anschließender empirischer Auswertung im Sommersemester 2015 geplant. Lehrende, die ebenfalls an einem Einsatz interessiert sind, sind eingeladen, mit Unterstützung des Autors den Einsatz in eigenen Lehrveranstaltungen anzugehen.

Literatur

- Akilli, Göknur (2007): „Games and Simulations: A New Approach in Education?“. In: *Games and Simulations in Online Learning: Research and Development Frameworks*. Information Science Pub., S. 1–20.
- Alistair Cockburn (2007): „A Detailed Critique of the SEMAT Initiative“. *Alistair.Cockburn.us*. Abgerufen am 14.10.2014 von <http://alistair.cockburn.us/A+Detailed+Critique+of+the+SEMAT+Initiative>.
- Breuer, Johannes (2010): „Spielend lernen? Eine Bestandsaufnahme zum (Digital) Game-Based Learning“. Landesanstalt für Medien Nordrhein-Westfalen (LfM).
- Fowler, Martin (2010): „Semat“. *martinfowler.com*. Abgerufen am 14.10.2014 von <http://martinfowler.com/bliki/Semat.html>.
- Gee, James Paul (2007): *What Video Games Have to Teach Us About Learning and Literacy*. Palgrave Macmillan.
- Gesellschaft für; Gesellschaft für Informatik e.V. (GI) (2005): „Empfehlungen der Gesellschaft für Informatik e.V. (GI) für Bachelor und Masterprogramme im Studienfach Informatik“.
- International Organization for Standardisation (ISO) (2007): „ISO/IEC 24744:2007 - Software Engineering -- Metamodel for Development Methodologies“.
- Ivar Jacobson; Bertrand Meyer; Richard Soley (2010): „Some critiques of the Semat initiative | SEMAT blog“. *SEMAT blog*. Abgerufen am 14.10.2014 von <http://sematblog.wordpress.com/2010/04/24/some-critiques-of-the-semat-initiative/>.
- Ivar Jacobson International (2014): „Developing and Customizing Practices | EssWork Practice Workbench“. Abgerufen am 25.06.2014 von http://www.ivarjacobson.com/EssWork_Practice_Workbench/.
- Jacobson, Ivar; Ng, Pan-Wei; McMahon, Paul; u. a. (2012): „The Essence of Software Engineering: The SEMAT Kernel“. In: *Queue*. 10 (10), S. 40:40–40:51, DOI: 10.1145/2381996.2389616.
- Jacobson, Ivar; Ng, Pan-Wei; McMahon, Paul E.; u. a. (2013): *The Essence of Software Engineering: Applying the SEMAT Kernel*. Upper Saddle River, NJ: Addison-Wesley Professional.
- Jöran Pieper (2013): „Alles nur Spielerei? Neue Ansätze für digitales spielbasiertes Lernen von Softwareprozessen“. In: Andreas Spillner; Horst Lichter (Hrsg.) *Tagungsband des 13. Workshops „Software Engineering im Unterricht der Hochschulen“ 2013*. Aachen: CEUR Workshop Proceedings, S. 131–139.
- Kahoot! AS (2014): „Kahoot! | Game-based blended learning & classroom response system“. Abgerufen am 30.10.2014 von <https://getkahoot.com/>.
- Kritzenberger, Huberta (2005): *Multimediale und interaktive Lernräume*. München : Oldenbourg.
- Ludewig, Jochen (2009): „Erfahrungen bei der Lehre des Software Engineering“. In: Jaeger, Ulrike; Schneider, Kurt (Hrsg.) *Tagungsband des 11. Workshops „Software Engineering im Unterricht der Hochschulen“ 2009*. Heidelberg: dpunkt.verlag, S. 75–86.
- Object Management Group (2014): „Kernel and Language for Software Engineering Methods (Essence) Version 1.0“.
- Object Management Group (2008): „OMG - Software & Systems Process Engineering Meta-Model (SPEM), v2.0“. Abgerufen am 26.10.2011 von <http://www.omg.org/cgi-bin/doc?formal/2008-04-01>.
- Paul E. McMahon (2013): „Essence: Why do we need it?“. *SEMAT blog*.
- Prensky, Marc (2007): *Digital game-based learning*. Paragon House.
- Reich, Kersten (2008): *Konstruktivistische Didaktik: Lehr- und Studienbuch mit Methodenpool*. Beltz.
- Rieber, Lloyd P. (1996): „Seriously considering play: Designing interactive learning environments based on the blending of microworlds, simulations, and games“. In: *Educational Technology Research and Development*. 44 (2), S. 43–58, DOI: 10.1007/BF02300540.
- SEMAT (2014): „Software Engineering Method and Theory“. Abgerufen am 21.10.2014 von <http://semat.org/>.
- Striewe, Michael; Goedicke, Michael (2013): „Modellierung und Enactment mit ESSENCE“. In: *Proceedings of Workshop „MVF - Modellierung von Vorgehensmodellen - Paradigmen, Sprachen, Tools“*, *Software Engineering 2013*. Aachen, Germany, S. 405–414.
- Wangenheim, Christiane Gresse von; Shull, Forrest (2009): „To Game or Not to Game?“. *IEEE Software*. 26 (2), S. 92–94.