Mehr Transparenz wagen: Mit Offenheit zum Erfolg

Wer kennt es nicht: explodierende Aufwände, instabile Konzepte, ein komplexes Umfeld, hohe Qualitätsansprüche. Herausforderungen, denen es zu begegnen gilt, um dem Kunden ein gutes Produkt zu liefern. Doch wie lassen sich solche Probleme beheben oder – im ersten Schritt – erkennen? Lösungsansätze diesbezüglich sind nicht immer einfach zu finden und scheitern nicht selten an der Unternehmenskultur. Basierend auf Projekterfahrungen präsentiert dieser Artikel drei Methoden, mit deren Hilfe sich die Transparenz des Software-Entwicklungsprozesses und der Arbeit der Entwicklungsteams deutlich steigern lässt. Mit dieser Transparenz können Herausforderungen besser erkannt und gezielter bewältigt werden. Gleichzeitig wirkt sie sich positiv auf die Unternehmenskultur aus.

Schlägt man das Wort „Transparenz“ im Duden nach, erhält man als Bedeutung unter anderem „Durchschaubarkeit“ und „Nachvollziehbarkeit“. Weiter bietet der Duden Synonyme wie „Aufrichtigkeit“, „Ehrlichkeit“, „Geradheit“, „Geradlinigkeit“ und „Offenheit“ an (vgl. [Dud15]). Klingt das nach Werten aus Ihrem Projektalltag?

Die Forderung nach Transparenz

Fakt ist, dass Transparenz insbesondere in der agilen Softwareentwicklung fest verankert ist, zumindest in der Theorie. Der offizielle Scrum Guide gründet die Grundidee von Scrum auf der Theorie empirischer Prozesssteuerung. Deren drei Säulen sind:

- Transparenz
- Überprüfung (*Inspect*)
- Anpassung *(Adapt)*

Dabei spielt Transparenz die tragende Rolle (vgl. [Sch13]). Nur wenn der Prozess und seine Elemente transparent sind, ist eine empirische Beobachtung möglich und die notwendigen Informationen stehen zur Verfügung, um den Prozess überprüfen und anpassen zu können.

Dies gilt nicht nur für Scrum. Agile Methoden und Prinzipien leben vom Gedanken des *Inspect & Adapts*, der Reflexion von Iterationen und der kontinuierlichen Verbesserung. So legt David J. Anderson für Kanban die Optimierung des Prozesses als Hauptziel fest. Um dies zu erreichen, nennt er als eines der Unterziele das „Herstellen von Transparenz über die Gestaltung und den Einsatz des Systems“ (vgl. [And11]). Kanban macht nicht nur nachvollziehbar, auf welchem Stand welche Aufgabe gerade ist und wann sie erledigt sein wird. Kanban lässt auch die Leistung des Teams erkennen. Außerdem werden der Prozess und seine Funktionsweise visualisiert mit dem magischen Effekt, dass die Auswirkungen des Handelns innerhalb des Systems für alle erkennbar werden (vgl. [And11]). Dies zeigt, wie Transparenz nicht nur den Prozess, sondern auch das gesamte Projektumfeld, inklusive der Beziehung zum Kunden und zum Management verbessern kann. Die Verbesserung basiert unter anderem auch darauf, dass mit der Transparenz Informationen in einer neuen Qualität zur Verfügung stehen. So nennt Mike Burrows als dessen Daseinsberechtigung „Menschen zu helfen, gute Entscheidungen zu treffen“ – und das nicht nur retrospektiv, sondern auch „bei der täglichen Arbeit“ (vgl. [Bur14]).

Transparenz ohne Grenzen?

Doch wie weit sollte man mit dieser Transparenz gehen? Gibt es eine Grenze? Im Zusammenhang mit dem agilen Kernwert „Individuals and Interactions“ fordert Jeff Sutherland „transparency of all data, actions and decisions” (vgl. [Sut15]). Ist eine derart allumfassende Transparenz im Projektalltag überhaupt möglich? Sutherland bringt hierbei eine wesentliche Herausforderung auf den Punkt: „Most teams avoid truth, transparency, and trust because of cultural norms or past negative experiences from conflict that was generated by honest communications”. Die Schaffung von durchschaubaren, nachvollziehbaren, ehrlichen Prozessen und Kommunikation ist nicht nur durch den Wunsch und das Engagement eines Teams erreichbar; sie fordert ein unternehmensweites Engagement und eine entsprechende Kultur. Letzteres ist mit Sicherheit die größere Herausforderung.

Folgen mangelnder Transparenz

Mit welchen Folgen ist nun bei mangelnder Transparenz zu rechnen? Die generische Antwort wäre aus Sicht der Agilität: Der Kunde bekommt nicht das, was er braucht, oder die Umsetzung des Kundenwunsches bedeutet für den Auftragnehmer erhöhte Kosten oder einen erhöhten Ressourcenaufwand. Es kann sein, dass die Aufwände gestiegen sind, weil dem Team Risiken bekannt waren, dem Management jedoch nicht frühzeitig mitgeteilt wurden. Auch kann es sein, dass mehr mit Annahmen gearbeitet wird als mit konkreten Anforderungen, weil es Probleme mit der Kommunikation zwischen Product Owner und dem Team gibt. Solche und eine Vielzahl weiterer Ursachen haben zur Folge, dass zu spät, in unzureichender Qualität oder nicht im benötigten Funktionsumfang geliefert wird. Dabei lassen sich die Ursachen im gesamten Unternehmen finden.

Genauso vielfältig wie die Folgen sind demnach auch die Lösungsansätze. Ein generelles Kochrezept gibt es nicht, zumal die bereits angesprochene Unternehmenskultur eine wichtige Rolle spielt. Die Schaffung der geforderten allumfassenden Transparenz kann da im ersten Schritt möglicherweise hinderlich sein. Tatsächlich kann aber gerade Transparenz, zu Beginn wohldosiert, dabei helfen, die vorliegenden Herausforderungen aufzudecken. Gleichzeitig lässt sie sich gezielt einsetzen, um die bestehenden Probleme zu lösen.

Es folgen drei praxiserprobte Methoden, die durch ein Mehr an Transparenz dabei unterstützen können, Herausforderungen zu identifizieren und zu beseitigen. Dabei fangen all diese Methoden im Team an, bieten aber zugleich ein hohes Potenzial, den Wert der Transparenz in das gesamte Unternehmen zu tragen.

Methode 1: Taskboard „Smells”

Anderson nennt „Visualize your flow“ („Visualisiere den Fluss der Arbeit“) als

erste Kerneigenschaft von Kanban (vgl. [And11]). Unabhängig von der Methode hat sich die Visualisierung des Arbeitsflusses, zum Beispiel in Form eines Taskboards, in agilen Vorgehensweisen bereits etabliert. Ob digital oder analog – Projektbeteiligte können am Board den Status und den Fluss der Arbeit visuell erfassen. Das Board strukturiert den Prozess und hilft dabei, den Überblick über die aktuellen Arbeiten zu behalten.

Das Taskboard kann demnach die Transparenz des Prozesses deutlich steigern. Das Wort „kann" ist hier jedoch bewusst gewählt. Erfahrungsgemäß fehlen in der Praxis oftmals wichtige Prozessschritte. Wie kann es dazu kommen?

Die einfachste Erklärung ist sicherlich die menschliche Fehlbarkeit: Schritte werden vergessen oder schlichtweg nicht erkannt. Eine andere Ursache: Das Team orientiert sich zu stark an der Fachliteratur und übernimmt von dort Beispiele ohne retrospektive Überlegungen und ohne entsprechende Anpassungen. Nicht so offensichtlich ist eine weitere Ursache, die in den unterschiedlichen Betrachtungskontexten der verschiedenen agilen Vorgehensmodelle zu finden ist: So zielt Kanban beispielsweise auf einen konstanten Fluss ab. Deshalb ist es sinnvoll, bei der Erstellung des Taskboards wirklich alle Prozessschritte, möglichst sogar teamübergreifend, in Betracht zu ziehen, um Engpässe oder eine zu geringe Auslastung identifizierbar zu machen.

Scrum hingegen konzentriert sich auf die Arbeit, die im aktuellen Sprint noch zu erledigen ist. Man bleibt im Teamkontext, weshalb die Ausgestaltung des Taskboards weniger Varianten zulässt und dies initial auch nicht gefordert wird. Man startet hier eher mit den Empfehlungen aus der Literatur und macht sich erst später Gedanken über Optimierungen. Daher fehlen unter Umständen von Anfang an wichtige Schritte.

Welche Prozessschritte sollten nun am Board visualisiert werden? Da sich die Prozesse von Team zu Team unterscheiden, ist diese Frage nicht einfach zu beantworten. Speziell die Frage nach dem Detailgrad ist immer eine Herausforderung. Man könnte sämtliche Prozessschritte bis ins Kleinste komplett abbilden, wobei man dann potenziell versucht ist, sich in Details zu verlieren. Bei einer zu groben Betrachtung hingegen fallen wichtige Schritte unter den Tisch. Die Projekterfahrung zeigt jedoch, dass es Indikatoren gibt, mit denen man Lücken am Board identifizieren kann. Diese Hinweise werden im weiteren Verlauf *Smells* genannt – angelehnt an die Metapher „Bad Smells in Code" von Kent Beck, die auf eine unsaubere Code-Stelle bei der Programmierung hinweist (vgl. [Fow00]). Prüft man das Board auf diese Smells, zeigt sich, ob dieses den für ein erfolgreiches Arbeiten notwendigen Umfang an Prozessschritten und Informationen visualisiert. Im Folgenden werden einige wichtige Smells mit ihren Indizien, möglichen Auswirkungen auf den Prozess sowie mögliche Lösungswege beschrieben.

- *Smell 1: Kommunikationslücken:* Reden Teammitglieder häufig aneinander vorbei? Gibt es insbesondere an den Übergängen der Prozessschritte Unklarheiten? Kommt es zu Reibungsverlusten an den Schnittstellen zu anderen Teams? Das Board ist eine Vereinbarung über den Prozess, die innerhalb des Teams und auch teamübergreifend offen kommuniziert ist. Bestehen im Team Unklarheiten zum Prozessablauf, wirkt sich das auf den Arbeits- und Informationsfluss aus. Ursache für diese Unklarheiten ist häufig die mangelnde Verfügbarkeit notwendiger Informationen. Visualisiert man diese am Board, ist sie für alle sichtbar und schließt nach und nach nicht nur die Lücke am Board, sondern auch im gelebten Prozess.
- *Smell 2: Unvollständige Arbeitspakete:* Beklagt sich das Team über unklare Konzepte? Findet der Test übermäßig viele Fehler? Kommen Arbeitsergebnisse nicht im nötigen Umfang im nächsten Prozessschritt an, dann kann es sein, dass die Übergänge inkorrekt oder unklar definiert sind. Das Board hat nicht nur die Aufgabe, die Prozessschritte zu visualisieren, sondern stellt auch die Übergänge sicher, zum Beispiel in Form einer *Definition of Done*.
- *Smell 3: Verdächtige Aufwandszahlen:* Ist die Team-Velocity nicht nachvollziehbar? Kämpft das Team mit hohen Zusatzaufwänden oder Storys, die in ihren Aufwänden explodieren? Ein „Ja" ist unter Umständen ein Hinweis auf implizite Tätigkeiten des Teams. Diese können durchaus notwendig sein. Sind sie aber nicht am Board sichtbar, lassen sich die Aufwände diesbezüglich schwer zuordnen und verschwinden in einem „schwarzen Loch", das gleichzeitig neue Herausforderungen verdeckt. Eine Betrachtung und Anpassung am Board hilft, Verständnis zu schaffen. So kann einerseits allen notwendigen Schritten ein entsprechender Stellenwert eingeräumt werden, andererseits lassen sich unnötige Schritte beseitigen und somit Aufwände reduzieren.
- *Smell 4: Vernachlässigung wichtiger Prozessschritte:* Vergisst das Team des Öfteren die Durchführung der Code-Reviews? Schlüpfen ab und zu ungetestete Tasks in das Release? Wenn Prozessschritte am Board fehlen, werden sie vernachlässigt, nur unbewusst durchgeführt, nicht in der notwendigen Sorgfalt oder gar nicht erst ausgeführt. Der Prozess ist nicht nur am Board lückenhaft, sondern auch in der Ausführung. Und dies ist den Projektbeteiligten vielleicht nicht einmal bewusst.
- *Smell 5: Ständige Rechtfertigung für Aufwände:* Muss sich das Team regelmäßig für die Aufwände bestimmter Tätigkeiten, zum Beispiel dem Grooming, vor dem Management rechtfertigen? Wenn Prozessschritte am Board fehlen und sie trotzdem durchgeführt werden, gelten sie schnell als Nebentätigkeit. Die Stakeholder bringen ihnen nicht den nötigen Respekt entgegen. Als Ergebnis findet man häufig lückenhafte Ausführungen und Schwierigkeiten, diese Schritte in den Arbeitsalltag einzuplanen. Diese Tätigkeiten müssen unter Umständen explizit als Schritt am Board aufgenommen werden. Hierdurch wird allen Beteiligten die Notwendigkeit der Tätigkeit noch deutlicher.

Ziel ist es, das Board und den gelebten Prozess soweit aneinander anzupassen, dass das Board die Projektrealität widerspiegelt. Dann erst entfaltet das Taskboard eine unglaubliche Effizienz. Die geschaffene Transparenz hilft, Herausforderungen deutlich schneller zu entdecken. Mit ihr versteht jeder Projektteilnehmer besser, welche Verantwortung er und die anderen Projektteilnehmer am Prozess tragen.

Ein konkretes Beispiel untermauert diese Punkte: Ein Entwicklungsteam beschwerte sich über explodierende Aufwände. Regelmäßig wurde der Sprint dadurch belastet und Storys wurden nicht fertig. Smell 3 hätte hier bereits den Blick auf das Taskboard lenken können. In diesem Fall handelte es sich aber nicht um eine versteckte Tätigkeit, die den Aufwand direkt erhöhte. Stattdessen wurden als Ursachen unreife Konzepte und zu spät erkannte technische Widrigkeiten identifiziert, was eher nach Smell 2 – also

Abb.1: Transparenz aller Prozessschritte am Board.

unfertigen Arbeitspaketen – roch. Ein Abgleich des Prozesses mit dem Board war in jedem Fall nötig und zeigte deutlich die Wurzel der Probleme. Das Board bildete den standardmäßigen Workflow „*ToDo – InWork – Ready4Test – Done*" ab. Doch wo kamen die Arbeitspakete ursprünglich her?

Im ersten Schritt belieferte das Produktmanagement des Unternehmens den Product Owner mit Konzepten. Soweit befand man sich noch außerhalb des Entwicklungsteams. Das Team prüfte anschließend die fertigen Konzepte in Form eines Groomings und erstellte daraus Arbeitspakete. Nachdem das Team die Arbeitspakete geschätzt und der Product Owner sie begutachtet hatte, wurden zu realisierende Arbeitspakete in die Spalte „ToDo" einsortiert. Dann setzte sich der Prozess wie am Board beschrieben fort.

Angeregt durch Smell 2 machte sich das Team Gedanken über die Schnittstelle zum Produktmanagement. Es überlegte sich, welche Forderungen an diese Schnittstelle bestanden. Neben einer angemessenen Anzahl an Konzepten, um den Sprint zu füllen, wurde ein Fokus auf die Qualität der Konzepte gelegt. Mit einer *Definition of Ready*, analog zur *Definition of Done*, schärfte das Team die Schnittstelle. Auch visualisierte das Team die Schnittstelle zusammen mit einer groben Abschätzung über die Anzahl und Qualität der Konzepte in Form einer Ampel am Board, um die Kommunikation mit dem Produktmanagement anzuregen, also indirekt Smell 1 vorzubeugen.

Im nächsten Schritt stellte das Team fest, dass insbesondere der Schritt des Groomings am Board nicht zu finden war. Zurückblickend fiel dem Team auf, dass das Grooming tatsächlich eher stiefmütterlich behandelt worden war. Die eigentlichen Arbeitspakete mussten parallel erledigt werden, das Grooming an sich wurde als zwischendurch zu erledigender Mehraufwand betrachtet. Man hatte sich nicht die nötige Zeit genommen, um die Konzepte auf Funktionalität und technische Machbarkeit zu prüfen. Abschätzungen wurden vage. Die in Smell 4 dargestellten Folgen eines fehlenden Prozessschrittes am Board waren deutlich sichtbar. Das Team entschloss sich also, den Grooming-Schritt in das Board zu integrieren. Durch die neue Transparenz am Board wurde das Grooming nun explizit und damit ein offizieller Arbeitsauftrag, für den Zeit vorhanden war und der im täglichen Arbeitsprozess Platz fand.

Das Ergebnis der Überarbeitung des Boards war erhöhte Transparenz (**siehe Abbildung 1**). Nicht nur die Arbeitsschritte des Teams konnte man nun vollständig beobachten, sondern auch die Herausforderungen bezüglich der Schnittstelle. Mit dem expliziten Grooming verbesserte sich die Qualität der Abschätzung der Storys, was zu einer drastischen Senkung der Anzahl explodierender Storys führte. Das Team konnte nun präzisere Vorhersagen treffen, die Planungssicherheit des Product Owners wurde gesteigert. Mit der geschärften Schnittstelle wurde im Laufe der Iterationen ein weiteres Problem sichtbar: Es stellte sich heraus, dass die Lieferung der Konzepte sowie deren Qualität schlechter waren als gedacht. Nun ließ sich erkennen, dass auch unreife Konzepte mangels Alternativen einfach in den Sprint „gekippt" worden waren.

Abschließend ein Hinweis zur Benutzbarkeit und Übersicht des Boards: In **Abbildung 1** ist das Board aus dem konkreten Beispiel zu sehen. Jedes Team hat jedoch einen anderen Prozess und so kann das Board in der Darstellung verschiedenartige Herausforderungen mit sich bringen. So wurde im Beispielprojekt nur auf der Ebene der User-Story gearbeitet. Wenn der Detailierungsgrad aber die Task-Ebene für sinnvoll erachten würde, steigt potenziell

Abb. 2: Blick von oben auf die Wege über das Eis.

die Anzahl der Zettel: Es wird unübersichtlicher. Hier kann man zum Beispiel auf Farb-Codes zurückgreifen, die die Zusammengehörigkeit zu den Storys visualisieren. Oder man verwendet *SwimLanes* je Story oder Feature. Sollte die Anzahl der Prozessschritte die des Beispiels überschreiten, reicht die Schrankwand eventuell nicht aus. In der Praxis sieht man häufiger Boards, die mehrere Meter Wand zieren. Um hier die Darstellung noch übersichtlich zu halten, empfiehlt es sich, Spalten konsequent für nur einen Prozessschritt zu nutzen und nicht wie in **Abbildung 1** aus Platzgründen zusammenzufassen.

Methode 2: Bitte anschnallen, wir heben ab

Im schwedischen See Storsjön befindet sich die Insel Fröson. Über eine Brücke ist die Insel mit dem Festland verbunden. Im Winter ist der See zugefroren. Das Eis wird dann so präpariert, dass Fußgänger und Autos den See überqueren können (**siehe Abbildung 2**). Damit sind eine interessante Abwechslung zur Brücke und kürzere Wege geboten. Ein Blick aus dem Flugzeug macht diese Wege sichtbar.

Auch im Projekt lohnt sich ein ähnlicher Blick von oben oder der bekannte „Schritt zurück“, um neue Wege zu entdecken. Im Sinne der kontinuierlichen Verbesserung ist in den agilen Vorgehensmodellen und Methoden, insbesondere in Scrum, der retrospektive Blick auf die Geschehnisse und Herausforderungen der letzten Iteration fest verankert. Es gibt jedoch Herausforderungen, die in Sprint-Retrospektiven nur schwer zu behandeln sind:

- Manche Probleme werden durch Betriebsblindheit, vor allem im laufenden Projekt, häufig übersehen. Erst ein gewisser Abstand, weg von den Projektgeschehnissen, lässt das Problem erkennen. Dies ist schwierig, wenn die Retrospektive direkt mit der Iteration verbunden ist.
- Es gibt Probleme, die nur schleichend und in gewissen Abständen erkennbar werden. Erst ein Trend über einen längeren Zeitraum macht sie sichtbar.
- Selbst wenn das Problem identifiziert ist, kann es schwer sein, dieses im laufenden Projekt zu beheben. Standardisierte Abläufe, Abhängigkeiten oder Politik können manchmal die Handlungsfähigkeit einschränken. Große Schritte wagt man vielleicht nicht, um das Produkt nicht zu gefährden.

Projekte haben gezeigt, dass eine zusätzliche Form der Retrospektive diese Herausforderungen angehen kann: der Blick von ganz oben. Ähnlich einem Flugzeug auf 10.000 Metern Reiseflughöhe betrachtet die „10.000-Meter-Retrospektive“ das Projekt aus größerer Distanz. Diese Retrospektive findet in deutlich größeren Intervallen statt als die Sprint-Retrospektiven, so zum Beispiel nach einer bestimmten Anzahl von Iterationen, zur Projekthalbzeit oder auch zum Projektabschluss für potenzielle Folgeversionen. Der größere zeitliche Abstand soll dem Team einen Blick von oben gewähren und somit mehr Transparenz verschaffen. Man sieht zwar unter Umständen weniger die Details, hat dafür aber einen Überblick, der die Entwicklung verschiedener Aspekte über längere Zeiträume zeigen kann. Auch Zusammenhänge zwischen verschiedenen Aspekten lassen sich besser herstellen. Das Team kann zudem aktiv einen anderen Blickwinkel einnehmen, der hilft, die Betriebsblindheit zu überwinden. Aus dieser neuen Perspektive können neue Erkenntnisse zu Problemstellungen generiert werden – Erkenntnisse, die dabei helfen, die Probleme besser zu verstehen und Lösungswege zu finden. Zudem steigern sie die Handlungsfähigkeit, indem größere Anpassungen am Prozess gegenüber möglichen Gegnern besser begründet werden können.

Eine der größten Herausforderungen ist die Moderation dieses Meetings. Die Betrachtung des längeren Zeitraums muss gut aufbereitet sein, um dem menschlichen Vergessen entgegenzuwirken. Ziel ist es, das Team auf einen Rundflug über die Geschehnisse der zu betrachtenden Iterationen mitzunehmen. Interessant dabei sind Erfolge, erkannte und behobene Hindernisse, besondere Ereignisse, die Entwicklung des Produkts und sonstige Statistiken. Die Informationen dazu lassen sich aus den Prozessartefakten ziehen, insbesondere aus den Sprint-Retrospektiven. Das bedeutet natürlich einiges an Vorbereitung, aber je besser das Team in die Vergangenheit eintaucht, desto mehr Erkenntnisse können aus der 10.000-Meter-Retrospektive gezogen werden.

OBJEKTspektrum ist eine Fachpublikation des Verlags:
SIGS DATACOM GmbH · Lindlaustraße 2c · 53842 Troisdorf
Tel.: 02241/2341-100 · Fax: 02241/2341-199
E-mail: info@sigs-datacom.de
www.objektspektrum.de
www.sigs.de/publications/aboservice.htm
SIGS DATACOM
FACHINFORMATIONEN FÜR IT-PROFESSIONALS

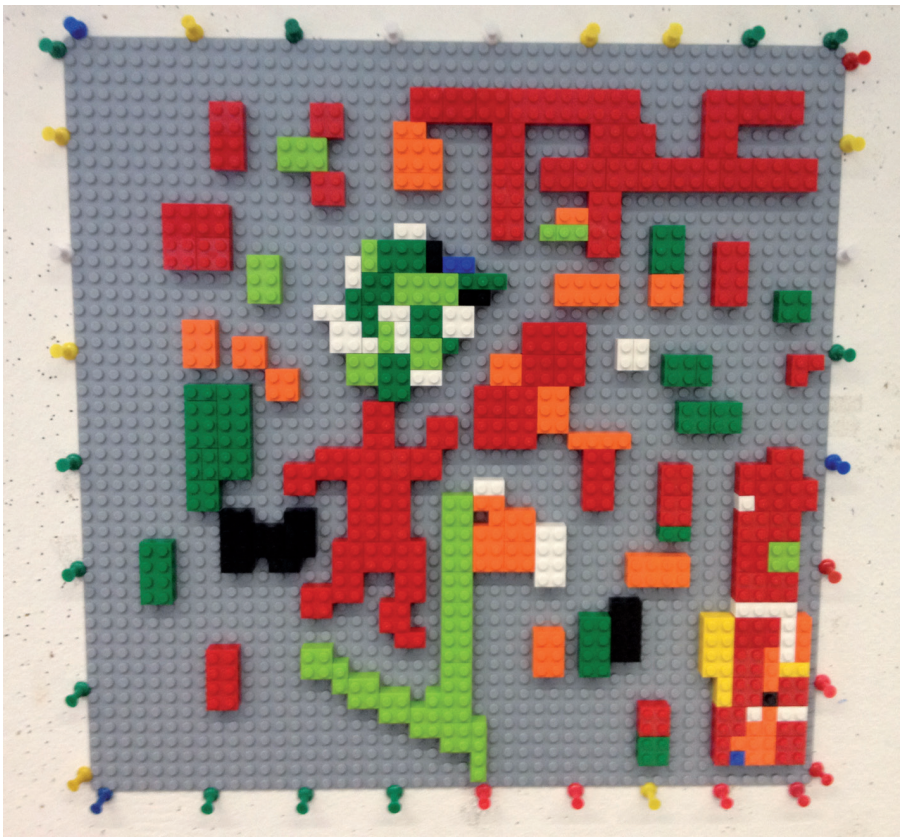

Abb. 3: Aufwände aus Lego-Steinen, bunt gemischt.

Erfahrungen haben gezeigt, dass für die erste Phase des Meetings, den Rundflug, etwas mehr Zeit und eine gewisse Offenheit eingeplant werden können. Die Teammitglieder sollen sich im lockeren Gespräch über ihre Erinnerungen und Empfindungen austauschen. Im nächsten Schritt werden Diskussionspunkte gesammelt, die das Team anschließend priorisiert bespricht. Dafür sind unter Umständen mehrere Sitzungen notwendig. Am Ende sollte natürlich ein Aktionsplan mit Verantwortlichen entstanden sein.

Kritischen Rufen über die hierbei zusätzlich generierten Zeitaufwände kann der potenzielle Nutzen gegenüber gestellt werden. Erfahrungsgemäß beläuft sich die Gesamtdauer, gestaffelt über kleinere Meetings, auf maximal einen Tag. Der Nutzen kann aber enorm sein, sodass die Zeit gut investiert ist. So ist die Herausforderung aus Methode 1 in solch einer 10.000-Meter-Retrospektive zum Vorschein gekommen, da die Prozesslücke erst mit etwas Abstand erkannt wurde.

Wichtig: Die 10.000-Meter-Retrospektive ist keine Alternative zur Sprint-Retrospektive, sondern eine Ergänzung.

Die bisherigen Erfahrungen mit dieser Art von Retrospektive beziehen sich auf den gleichen Teilnehmerkreis wie bei den normalen Iterationsretrospektiven. Eine interessante Ergänzung beschreiben Esther Derby und Diana Larsen (vgl. [Der06]). Die Autorinnen versuchen hier, den besprochenen Weitblick und den Perspektivwechsel dadurch zu erreichen, dass Projektbeteiligte außerhalb des Projektteams, zum Beispiel das Management, Marketing oder Sales, ebenfalls zum Meeting eingeladen werden. Dabei verfolgen sie das Ziel, mithilfe der unterschiedlichen Blickwinkel auf den Gesamtprozess einen unternehmensweiten Lernprozess zu ermöglichen. Allerdings rücken hier ganz andere Dinge in den Fokus als bei der 10.000-Meter-Retrospektive: Teaminterne Prozesse oder Herausforderungen stehen weniger im Vordergrund als der unternehmensweite Kontext. Eine komplette Vermischung wäre weniger erstrebenswert, sinnvoll sind getrennte Meetings. Aber eine wohlüberlegte Einladung bestimmter Personen kann durchaus interessant sein, insbesondere wenn Herausforderungen zu einer bestimmten Schnittstelle existieren.

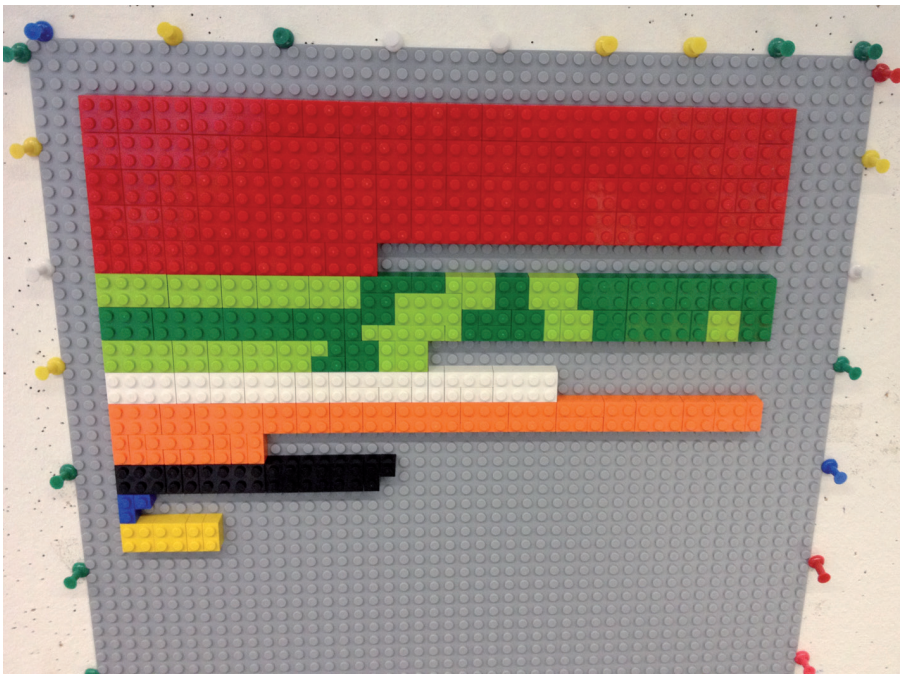

Abb. 4: Aufwände aus Lego-Steinen, sortiert.

Methode 3: Besseres Reporting mit Lego-Steinen

Es ist kein Geheimnis: Zahlen sind nur gut, wenn sie klar und einfach zu lesen und wenn sie leicht zu erheben sind. Sind sie nur schwer zu erheben, verfälscht sich die Statistik infolge unvollständiger Zahlen. Ist der Report unübersichtlich und schlecht strukturiert, wird er nicht gelesen und die Botschaft wird nicht übermittelt.

Ein Negativbeispiel aus einem Projekt veranschaulicht dies: Um die sinkende Velocity aufgrund steigender Zusatzaufwände zu verdeutlichen, entschloss sich das Team, eine Liste zu pflegen, auf der die Zusatzaufwände notiert, kategorisiert und aufsummiert wurden. Damit sollten die Quellen der Zusatzaufwände identifiziert und bei Bedarf an das Management kommuniziert werden. Die Liste hing für alle sichtbar aus. Bis auf einige wenige Einträge während des Sprints wurde die Liste jedoch meist erst zum Sprint-Ende auf Nachfrage des Scrum Masters gefüllt. Warum? Der Aufwand, diese Liste zu pflegen und die Stunden aufzuschreiben, war einfach zu lästig, als dass die Teammitglieder ihn mit Überzeugung getätigt hätten. Am Ende hing ein Blatt Papier mit vielen Zeilen handschriftlicher Einträge an der Wand, viel zu unattraktiv, um im Vorbeigehen einen Blick drauf zu werfen. Notgedrungen wurde die Liste am Ende des Sprints aus der Erinnerung gefüllt und erst anschließend analysiert, was den Erkenntnisgewinn natürlich reduzierte. Entsprechend den einleitenden Worten war dies ein kompletter Fehlschlag.

Bis der Scrum Master des Teams in einer Konferenzpause eine Idee aufschnappte, ein Stück aus der Kindheit vieler: Lego-Steine. Die Liste wurde durch eine Lego-Platte ersetzt und jeden Tag nach dem Stand-up setzten die Teammitglieder je gearbeiteter Stunde eine Lego-Noppe in verschiedenen Farben, einer Aufwandskategorie entsprechend. Über den Sprint hinweg entstand so ein Bild über die Zusatzaufwände (**siehe Abbildung 3 und Abbildung 4**).

Der Vorteil der Lego-Steine ist vielfältig. Das Setzen der Steine bzw. Aufwände macht Spaß und ist unkompliziert. Die Möglichkeit, ideenreiche Muster zu setzen, animiert und motiviert, fleißig mitzuwirken. Man erhält ein sehr vollständiges Bild über die Aufwände. Ebenso wichtig ist ein zweiter Aspekt: Aufmerksamkeit. Nur wenige Minuten nach Einführung der Steine stand der komplette Führungskreis staunend und bewundernd vor der Lego-Platte. Der Blickfänger hat bis heute seine Wirkung nicht verfehlt, die Kommunikation kommt automatisch zustande. Dies ist eine Stufe der Transparenz, die die vorherige Liste nicht erreicht hat.

Aber auch hier ist Vorsicht geboten. Das Lego-Board erhöht die Sichtbarkeit enorm: Nicht nur den Führungskreis, auch Projektfremde lockt der Spieltrieb. So kann sich das Team sehr schnell beobachtet fühlen. Im Zusammenhang mit dem Wert „Respekt“ geht Mike Burrows auf die eingangs erwähnte Frage ein, ob es zu viel Transparenz geben kann. Er sagt nicht, dass es eine Grenze gibt, antwortet aber mit einer Gegenfrage: „Ist es möglich Transparenz auf eine gedankenlose oder leichtsinnige Weise zu realisieren?“ (vgl. [Bur14]). Es ist also wichtig, den richtigen Grad an Transparenz abzuwägen. Im Beispielprojekt äußerte ein Teammitglied die Idee, die Steine personalisiert zu setzen. Das Feedback aus dem Team dazu war negativ. Abgesehen davon, dass als Grundgedanke stets das Team und nicht der Einzelne im Vordergrund stehen sollte, kam die Angst der Bloßstellung auf. Transparenz hat viel mit Vertrauen zu tun, das durch Gedankenlosigkeit und Leichtsinn aufs Spiel gesetzt werden kann. Der Bedarf an solch einem Vertrauen besteht nicht nur innerhalb eines Teams, sondern auch vom und zum gesamten Projektumfeld. Insbesondere hier kann eine Öffnung durch eine transparente und ehrliche Kommunikation ein Antriebsmotor für Veränderungen des Projektumfeldes und der Unternehmenskultur sein.

Wie geht man nun konkret vor? Zunächst sollte man sich darüber im Klaren sein, was kommuniziert werden soll. Dann muss, wie eingangs erwähnt, die Einfachheit der Erfassung gewährleistet sein. Um die Sichtbarkeit zu unterstützen, muss die Kommunikation positiv auffallen. Da sind der Kreativität keine Grenzen gesetzt. In Summe sollte auf allen Seiten der Spaßfaktor stimmen.

Lego-Steine sind hier nur ein Beispiel für eine lebendige, polarisierende, attraktive Art der Darstellung. Vielleicht wird nicht überall eine derartige Verspieltheit gern gesehen. Aus Erfahrungen lässt sich aber sagen, dass selbst in sehr konservativen Umfeldern eine solche Auflockerung akzeptiert wird. Und genau hierin liegt das Geheimnis.

Fazit

Die beschriebenen Methoden helfen dabei, die Transparenz in der agilen Softwareentwicklung zu erhöhen und somit Herausforderungen zu entdecken und Probleme zu beheben. Gleichzeitig werden Schritte getan, die langfristig zu einer unternehmensweiten Transparenz führen und die Unternehmenskultur verändern können. Egal ob durch eine Visualisierung der Schnittstellen, den erweiterten Blick von oben, das Einladen teamfremder Projektbeteiligter oder das effiziente Verteilen von Informationen: Transparenz führt Schritt für Schritt zum agilen Unternehmen und zum zufriedenen Kunden. ‖

Literatur & Links

[And11] D.J. Anderson, Kanban: Evolutionäres Change Management für IT-Organisationen, dpunkt.verlag, 2011

[Bur14] M. Burrows, Kanban from the inside, Blue Hole Press, 2014

[Der06] E. Derby, D. Larsen, Agile Retrospectives: Making Good Teams Great, The Pragmatic Bookshelf, 2006 (3. Auflage, Januar 2008)

[Dud15] Duden, Transparenz, 23.09.2015, siehe:
`www.duden.de/rechtschreibung/Transparenz`

[Fow00] M. Fowler, Refactoring, Addison-Wesley, 2000

[Sch13] K. Schwaber, J. Sutherland, Der Scrum Guide, 2013, siehe:
`http://www.scrumguides.org/docs/scrumguide/v1/Scrum-Guide-DE.pdf`

[Sut15] J. Sutherland, Agile Principles and Values, 2015, siehe:
`https://msdn.microsoft.com/en-us/library/dd997578%28v=vs.100%29.aspx`

Der Autor

‖ David Uhlenberg
(David.Uhlenberg@methodpark.de)
arbeitet als Software-Engineer bei Method Park im Team Medical Applications. In der Rolle des Scrum Masters betreut er vor allem Kunden aus der Medizintechnik bei der Einführung und Anwendung agiler Methoden und Praktiken.