



Kanban in der Produktentwicklung

Ist es ausreichend „agil“ zu sein?

Agile Ansätze in der Softwareentwicklung haben eine ganze Branche verändert, allen voran Scrum. Daneben hat sich die Kanban-Methodik zum zweiten agilen Standard gemauert. Gerade in Umgebungen, die sich mit dem revolutionären Ansatz von Scrum schwer tun, zum Beispiel Organisationen die Elektronik und Mechanik entwickeln oder große hierarchische Organisationen, sehen in Kanban einen einfachen Einstieg in die Agilität. Die Gefahr bei diesem evolutionären Ansatz ist jedoch, dass sich Unternehmen auf Zwischenschritten ausruhen und zu früh eine Bilanz über die Sinnhaftigkeit von agilen Ansätzen erstellen könnten. Dieses Dokument erklärt die Herausforderungen in der Produktentwicklung, die Lösungen die Kanban mitbringt und die Dinge, die noch getan werden müssen, um die Entwicklung weiter zu bringen - über die Agilität hinaus.

Ein neuer Blickwinkel

Motivation

Die Herausforderungen, die sich heute in der Produktentwicklung stellen, sind hinlänglich bekannt: Immer komplexere Produkte, immer kürzere Entwicklungszyklen. Insgesamt nimmt also die Dynamik in Entwicklungsprojekten immer mehr zu. Agile Ansätze begegnen diesen Herausforderungen durch den Einsatz bewährter Praktiken aus der Manufaktur und der Abkehr vom verbreiteten Taylorismus.

Klassisches Projektmanagement übernimmt Produktionsdenken und tayloristisches Gedankengut in die Welt der Entwicklung. Vor allem die klassischen Planungs- und Monitoring-Aspekte funktionieren dabei in der Produktentwicklung nur unterdurchschnittlich oder gar nicht. Der

Grund dafür sind verschiedene Wahrscheinlichkeitsverteilungen, sowohl bei der Ankunftsrate von neuen Aufgaben, wie auch bei der Schätzgenauigkeit, also der Bearbeitungszeit.

Bei diesen beiden Aspekten lebt uns die Produktion vor, wie mit der Verringerung von Variabilität, also dem Wirken hin zu schmalbandigen Verteilungen, der Prozess planbar und steuerbar wird.

Die Tragik dabei ist, dass das was Entwicklung ausmacht, nämlich Innovationsfähigkeit, von der vorhandenen Variabilität lebt. Wer versucht in der Entwicklung einen reproduzierbaren und planbaren Ablauf zu schaffen, wird zwangsläufig Innovation verhindern.

Doch wie kann mit diesem Umfeld, voller Innovation und Variabilität und den damit vorhandenen und notwendigen breiten Streuungen, sinnvoll umgegangen werden? „Sinnvoll“ bezieht sich hier auf: Ökonomie, Innovation und Motivation

„Wer versucht in der Entwicklung einen reproduzierbaren und planbaren Ablauf zu schaffen, wird zwangsläufig Innovation verhindern.“

der Mitarbeiter, wobei diese drei Punkte eigentlich zusammenhängen. Es lohnt sich für vorrübergehend mental der Fertigung den Rücken zu kehren

und die Frage zu stellen: „Wer hat gelernt mit unplanbarem Auftreten von Aufgaben und deren unplanbaren Bearbeitungszeiten umzugehen?“.

Umgeschaut

Es gibt eine Branche, die seit jeher mit den oben genannten Herausforderungen umgehen muss, die Telekommunikationsindustrie. Das Verhalten von Telefonie- und Internet-Kunden kann nicht geplant werden. Allenfalls kann über statistische und mathematische Modelle eine bestimmte Service-Qualität sichergestellt werden.

Aus diesen Modellen kann die sogenannte Warteschlangentheorie sehr gut für Entwicklungsprozesse herangezogen werden. Sie stellt mathematische Modelle für den Zusammenhang der Länge einer Warteschlange und der Auslastung des sogenannten Servers. Für den einfachen Fall, dass sowohl die Ankunftsrate von Aufgaben, wie auch deren Bearbeitungszeit Poisson-verteilt sind (das passt in der Entwicklung, wie in der Telekommunikation meist recht gut), ergibt sich folgender Zusammenhang:

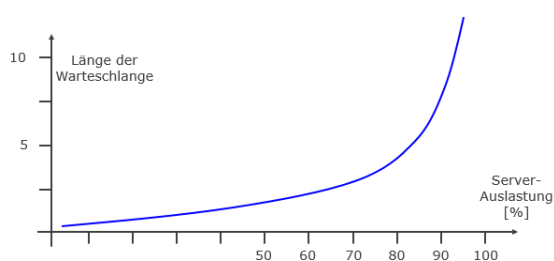


Abb. 1: Zusammenhang Auslastung und Warteschlange

Aus der qualitativen Darstellung ist ersichtlich, dass die Länge der Warteschlange mit der Auslastung des Servers exponentiell nach oben geht. Bei den viel zitierten 80% Auslastung wäre die Länge der Warteschlange lediglich 3,2 Aufgaben, bei 95% sind es schon 18 Aufgaben. Das alles sind Durchschnittswerte für den eingeschwungenen Zustand.

Sind diese Ansätze wirklich nützlich für die Produktentwicklung? Am eindeutigsten ist die Anwendbarkeit wenn wir als „Server“ interne Dienstleister betrachten, wie zum Beispiel Messlabore, Prüfstände oder Experten. Gerade hier wird in der Praxis viel Liegezeit erzeugt, obwohl nur kleine Erweiterungen der Server-Kapazität eine sehr große Reduzierung in der Wartezeit (diese ist proportional zur Länge der Warteschlange) erzielen können.

„Die wenigsten Unternehmen machen sich Gedanken über die Kosten von Liegezeiten in der Entwicklung. An dieser Stelle ist die Fertigung bereits seit 70 Jahren einen Schritt weiter.“

Dieser Zusammenhang erklärt auch den Einfluss von Schätzfehlern. Schätzfehler beeinflussen die Auslastung. Je nach Betriebspunkt hat diese Veränderung eine kleine Wirkung (Betriebspunkt bei kleinen Auslastungen) oder große Auswirkungen (bei großen Auslastungen). Dies ist aus den unterschiedlichen Steigungen der Kurve in den Betriebspunkten ersichtlich. In welchem Betriebspunkt Entwicklungsorganisationen meist betrieben werden, möge der Leser beurteilen.

Selbstverständlich gelten die Zusammenhänge auch für die „normale“ Entwicklungsmannschaft, auch wenn zuerst die Anwendung bei den internen Zulieferern ins Auge springt.

Umgedacht

Interessant ist die Auswirkung von Warteschlangen und damit Liegezeiten aus betriebswirtschaftlicher Sicht. Die wenigsten Unternehmen machen sich Gedanken über die Kosten von Liegezeiten (englisch: Cost of Delay, COD) in der Entwicklung.

An dieser Stelle ist die Fertigung bereits seit 70 Jahren einen Schritt weiter. Mit der Entwicklung des Toyota Produktionssystems (TPS) in den 1940er Jahren kam in der Produktion die Erkenntnis auf, dass nicht wie bisher angenommen die Betriebskosten der Maschinen und die Löhne der Arbeiter die Kostentreiber sind, sondern die Bestände an unfertigen Erzeugnissen in den Produktionshallen. Als Konsequenz wurde auf schnelle Durchlaufzeiten optimiert (Fluss) und dabei akzeptiert, dass Mensch und Maschine von Zeit zu Zeit nichts zu tun haben.

Mit dieser Erkenntnis hängt die Denkweise der Entwicklungsorganisationen der Fertigung noch hinterher, auch wenn zum Beispiel Kanban zum Beispiele hier nach vorne prescht, indem es Blockaden im Fluss zulässt und es somit hoffähig macht dass manche Entwickler nicht zu 100% ausgelastet sind. Ein Grund für das fehlende Umdenken ist die Herausforderung, die



Verzögerungskosten (COD) in der Entwicklung zu berechnen. Dies ist noch deutlich schwieriger als überhaupt Warteschlangen in der Entwicklung zu erkennen und zu messen. Genau hier kann die Agilität helfen, denn agile Backlogs sind visualisierte Warteschlangen. Es bleibt also die Aufgabe, mit den Verantwortlichen in den Dialog zu gehen um sich so an das Thema COD heranzutasten.

Wenn eine Organisation für den freien Markt produziert, können die COD zumindest grob abgeschätzt werden: „Wie viele Maschinen werden wir dieses Jahr weniger verkaufen, wenn wir einen Monat später auf den Markt kommen?“. Diese Frage ist der Einstieg in eine erste COD-Diskussion. Weniger transparent ist das Thema COD bei Auftragsentwicklungen wie z.B. bei Automobilzulieferern, doch auch hier sollte nach einer Lösung gesucht werden um COD greifbar zu machen.

Nur wer seine COD kennt, kann Maßnahmen in der Entwicklung (Werkzeuge, Technologien, Schulungen, Prozessverbesserungen, usw.) betriebswirtschaftlich begründen. Alles andere sind die bisher praktizierten Bauchentscheidungen, die nicht immer das wirtschaftliche Optimum darstellen und mangels Nachweisen oft angreifbar sind.

Auch Organisationen ohne bekannte COD können von der Warteschlangentheorie profitieren: Aus dem oben grafisch dargestellten Zusammenhang ergibt sich, dass zwei Betriebspunkte unabhängig von der Höhe der COD betriebswirtschaftlich indiskutabel sind. Sowohl 0% als auch 100% Auslastung der Entwickler.

Gutes Denken, schlechtes Denken

Auch wenn „Produktionsdenken“ bei der Projektplanung in Entwicklungsprojekten wenig hilfreich ist, lohnt es sich doch einen Blick in die Fabrikhalle zu werfen, wenn es darum geht den Durchsatz in der Entwicklung zu steigern, also den Fluss zu verbessern.

Als erstes sticht hier das Thema „Losgrößen“ ins Auge. Seit dem Toyota-Produktionssystem

und der daraus entstandenen Lean-Bewegung, ist dieses Thema eines der am meisten diskutierten Themen in der Fertigung.

Die mit dem TPS eingeführte Einzelstückfertigung „Single Piece Flow“ reduzierte die Durchlaufzeit um Zehnerpotenzen und steigerte in großer Masse die Flexibilität spät auf Änderungswünsche einzugehen. Grund genug nach Losgrößen in der Entwicklung zu suchen. Während sich die Software-Entwicklung dank (meist) geringer Querabhängigkeiten von Anforderungen mit „Single-Feature-Flow“ leicht tut, sind in Mechanik und Elektronik gewisse Losgrößen durch Produkt und Technologie gegeben. Interessant sind Losgrößen jedoch bei Entwicklungsartefakten. Hier ist zu beobachten, dass oft mit maximaler Losgröße in sogenannten „Stage-Gated“ Prozessen gefahren wird: Das Projekt wird gesamt von Stufe zu Stufe weitergeschaltet. Dadurch warten zum Beispiel viele Dokumente auf ein Review oder eine Weiterverarbeitung, weil noch nicht alle Dokumente der aktuellen Stufe fertiggestellt sind.

Dass hier große Verzögerungskosten (COD) entstehen, liegt auf der Hand. Auch hinsichtlich Risiko und Qualität sind große Lose ungünstig, das späte Feedback hat durch diese beiden Aspekte ebenfalls spürbare wirtschaftliche Auswirkungen. Kurz: Stage-Gated Prozesse erhöhen ohne Not Kosten und Risiko und können die Qualität negativ beeinflussen.

Eine weitere Methodik aus der Lean Production, welche der Verbesserung des Flusses dient, ist die Limitierung gleichzeitig geleisteter Arbeit (englisch: Work in Process, WIP). Für die Entwicklung bedeutet dies: Reduzierung von Multitasking. Man geht davon aus, dass die „Leistung“ eines Entwicklers um 20% je zusätzlicher Aufgabe abnimmt. Ein Entwickler, der in drei Projekten gleichzeitig arbeitet, kann also nur noch 60% seiner Arbeitszeit in das Unternehmen pro-

„Stage-Gated Prozesse erhöhen ohne Not Kosten und Risiko und können die Qualität negativ beeinflussen.“

duktiv einbringen. 40% gehen in Form von Reibung bei den notwendigen Kontextwechseln verloren (mentale Rüstkosten).

Dies ist ein durchaus bedeutender Faktor wenn es um das Thema Entwicklungsperformance geht. Scrum wie auch Kanban limitieren bewusst die gleichzeitig zu leistenden Aufgaben (WIP-Limitierung) nicht nur zur Freude der Entwickler, vor allem auch zur Freude der Manager, die die betriebswirtschaftliche Verantwortung tragen.

Ein weiteres prägnantes Thema im TPS sind dezentrale Entscheidungen. An welcher Stelle soll was entschieden werden? Je mehr Kompetenz „nach unten“ verlagert wird, desto schneller und damit kostengünstiger sind Entscheidungen. Agilität sieht sehr autonome Teams vor, jedoch muss ein Mechanismus gefunden werden, mit dessen Hilfe Entscheidungen nicht nur schnell, sondern auch in eine für die Organisation zielführende Richtung getroffen werden. Dezentrale Entscheidungen erfordern also zentrale Koordination, dies kann zum Beispiel über Regelwerke erfolgen. So spannend dieser Aspekt ist, für dieses Dokument sollen der dargestellte Rahmen genügen.

Kanban als Lösung?

Einmal wie oben beschrieben den Blickwinkel verändert, stellt sich die Frage: Wie können die aufgestellten Forderungen in der Praxis umgesetzt werden? Wer mit Kanban vertraut ist bemerkt, dass Kanban viele der angesprochenen Punkte löst bzw. lösen könnte. Doch der Reihe nach. Was bedeutet „Kanban in der Entwicklung“?

Historie eines Pull-Systems

Die Geschichte von „Kanban in der Entwicklung“ beginnt wider Erwarten nicht in der Produktion von Toyota, sondern in einem Entwicklungsteam bei Microsoft und mit einem israelischen Physiker.

David Anderson, Projektleiter in einem Wartungsprojekt bei Microsoft, hatte damit zu kämpfen, dass sein Team, wie eine überlastete Autobahn, nahezu am Stillstand operierte. Der Grund: zu viele gleichzeitige Aufgaben. Anderson versuchte daraufhin die bewährte Produktionssteuerung „Drum-Buffer-Rope“ (DBR) des israelischen Physikers Eliyahu Goldratt zu verwenden, indem er Materialfluss durch Aufgabenfluss ersetzte. Basis hierfür war die Visualisierung der Team-Aufgaben mit Haftnotizen auf einem Whiteboard. Die Umsetzung von DBR mit den Zettelchen in der Entwicklung gestaltete sich jedoch schwierig. Auf Empfehlung von Donald Reinert-

sen, stellte Anderson sein Whiteboard von DBR auf Kanban um, welches Jahrzehnte zuvor im Rahmen des TPS entstanden war. Damit gelang Anderson der Durchbruch und „Kanban in der Entwicklung“ war geboren. Es unterscheidet sich vom Produktions-Kanban dadurch, dass Anderson die Eng-

pass- und Pufferungskonzepte von DBR beibehielt und lediglich die Steuerung auf Kanban umstellte. Eine Ausführliche Erörterung der Vorteile von Kanban gegenüber DBR würde hier den Rahmen sprengen.

Die Lieferzeit für die Aufgaben in Andersons Team verringerte sich von 5 Monaten auf 2 Wochen, die Einhaltungquote des Zieldatums stieg von 0% auf 90%. Und dies alles innerhalb von 9 Monaten.

Für die detaillierte Darstellung des Kanban-Konzepts sei auf die entsprechende Literatur verwiesen.

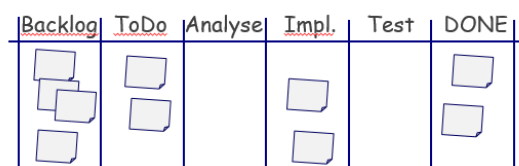


Abb. 2: Kanban-Board in der Entwicklung



Kanban ist wie DBR eine Pull-Steuerung: Es werden nur so viele Aufgaben ins System gezogen wie das System verkraften kann. Die Verantwortung des Managements verlagert sich vom Einsteuern (Push) und vom Micromanagement im Team hin zu der Priorisierung der Aufgaben. Zwei Aufgaben können nicht dieselbe Priorität haben. Nur so kann das Team die Aufgaben sequentiell abarbeiten und Multitasking vermieden werden.

Diese Umstellung kann für die Führungskräfte schwierig sein, die bisher immer alle Aufgaben gleichzeitig als gleich wichtig ins Team „gepusht“ haben. Sie müssen nun Entscheidungen zur Priorisierung treffen und diese in der Organisation vertreten.

Agiles Kanban

Kanban hat also nach erfolgreicher Einführung schon einige agile Themen gelöst. Zum einen bekommen Führungskräfte die wichtige (vielleicht ungewohnte) Verantwortung über Prioritäten zu entscheiden und überlassen den Arbeitsablauf vollständig dem Team. Zum anderen bietet Kanban (wie übrigens auch DBR) eine Limitierung des WIP, also der gleichzeitig angefassten Aufgaben. Damit bringt Kanban die Organisation schon deutlich weiter, ist aber noch nicht agil im eigentlichen Sinne. Reines Kanban könnte vielleicht als „Lean Development“ betrachtet werden, denn es übernimmt Steuerungsmechanismen aus der Produktion erfolgreich in die Entwicklung.

Agilität ist dagegen mehr als Priorisierung, Dezentrale Kontrolle und WIP-Limitierung. Agilität erwartet eine enge Zusammenarbeit mit dem Kunden des Projekts oder zumindest dessen Vertreter. Dieser Aspekt, bei Scrum zum Beispiel explizit in der Rolle des Product Owners dargestellt, muss bei einer Kanban-Implementierung bewusst „hineinmodelliert“ werden.

Die enge Zusammenarbeit mit dem Kunden ist jedoch nur gewinnbringend für das Projekt, wenn regelmäßig eine Auslieferung erfolgt. Damit kann hochwertiges Feedback über die Anforderungen und den Projektfortschritt erzeugt werden. Dies wiederum setzt die schleichende Akkumulierung von Abweichungen hinsichtlich Anforderungen, Zeitplan und Qualität regelmäßig zurück. Eine Serie vieler kleiner abgeschlossener Mini-Projekte, wenn man so will. Diese regelmäßige Auslieferung wird als Kadenz bezeichnet. Auch hier bietet Scrum eine fertige Lösung: Den Sprint.

Als letzter Punkt des Deltas zum agilen Vorgehen sei die Team-Arbeit angeführt. Das Kanban-Board bietet zunächst einmal lediglich ein paar Lean-Mechanismen und eine Visualisierung der Arbeit. Für eine Selbstorganisation des Teams ist diese Visualisierung jedoch der Schlüssel. Das ständig präsente Board verringert Kommunikati-

onsaufwand zur Koordination der gemeinsamen Arbeit und schafft somit Ressourcen für wichtige inhaltliche Kommunikation. Viele Organisationen übernehmen die bei Scrum bewährten täglichen Team-Meetings vor dem Board und das Retrospektiven-Meeting, in dem sich das Team regelmäßig über Verbesserungspotentiale austauscht.

Kanban soll leben, also sich laufend verändern und verbessern. Das Kanban-Board und ein mit entsprechenden Befugnissen und Vertrauensvorschuss ausgestattetes Team sind der Schlüssel dazu.

Für die volle Wirkung der Agilität ist eine Kanban-Einführung also noch nicht ausreichend, agile Elemente müssen bewusst in das Konzept mit aufgenommen werden. Für träge Organisationen bietet Kanban aber den Vorteil, dass Agilität in vielen kleinen Schritten eingeführt werden kann. Statt einer Revolution wie bei Scrum, kann hier eine Evolution (salopp: Salomitaktik) gewählt werden. Im Gegenzug muss mehr Aufwand in

„Kanban soll leben, also sich laufend verändern und verbessern. Das Kanban-Board und ein mit entsprechenden Befugnissen und Vertrauensvorschuss ausgestattetes Team sind der Schlüssel dazu.“



das Konzept und dessen Ausrollung gesteckt werden. Hierzu im Folgenden einige Hinweise:

Rollout

Kanban setzt im Gegensatz zu Scrum auf bestehenden Prozessen auf und belässt zunächst die vorhandenen Rollen und Verantwortlichkeiten. Dies erleichtert den Veränderungsprozess, der mit der Einführung von Agilität einhergeht erheblich.

Der erste Schritt wird immer eine rudimentäre Kanban-Implementierung sein: Der bestehende Prozess wird abstrahiert auf dem Board abgebildet und mit den Karten wird der Verlauf der Arbeit und gleichzeitig auch der WIP dargestellt. Oft beginnt die Einführung ohne harte WIP-Limits damit die tägliche Arbeit des Teams zunächst wie gewohnt weiter gehen kann. An dieser Stelle muss aber schon definiert werden, wer für die Pflege des Backlogs, der Eingangsliste mit den zu erledigen Aufgaben, zuständig ist: Welche Personen und Gremien steuern Aufgaben ein und wer entscheidet anhand von welchen Kriterien über die Priorisierung?

Das mit der Einführung vorhandene Pull-System kann seine Wirkung in diesem Moment noch nicht entfalten, es greift erst mit der Einführung von WIP-Limits. Wenn mit mehreren Teams gearbeitet wird, oder mit internen Zulieferern, wird in diesem ersten Schritt auch die Zusammenarbeit zwischen den verschiedenen Kanban-Boards und Backlogs festgelegt. Das alles ist noch kein agiler Paukenschlag, die Beteiligten gewöhnen sich jedoch an die Arbeit mit dem Board und Probleme im Konzept können erkannt und abgestellt werden. Ebenso können die Team-Meetings eingeführt, getestet und verbessert werden.

Der zweite Schritt sollte die Einführung und die Durchsetzung von WIP-Limits sein: Ein kleiner Schritt auf dem Kanban-Board, mit großer Auswirkung für die Organisation. Was bisher allen nur theoretisch klar war, nämlich dass das

Entwicklungsteam nicht viele Dingen gleichzeitig machen kann, manifestiert sich nun auf dem Kanban-Board. Aufgaben welche für die Organisation weniger wichtig sind als andere, werden hinten angestellt. Im Gegenzug dazu werden wichtigen Aufgaben in bisher ungeahntem Tempo fertiggestellt.

Der dritte Schritt sollte sich dem Thema „Kadenz“ widmen. Interessant ist dabei zunächst die Frage, was eine Auslieferung ist. Während in der Software eine Auslieferung meist ein Stück benutzbare Software ist (und diese auch in kurzen

Intervallen immer wieder ausgeliefert werden kann), stehen Elektronik und Mechanik vor dem Problem hoher Transaktionskosten für Musterbau und Test, sowie den Querabhängigkeiten zwischen den Anforderungen. Hier gilt es festzulegen, was denn eine Auslieferung sein kann. Von Konzepten über Simulationen bis hin zu physischen Mock-Ups kann die Palette reichen.

Für die weiteren Schritte bei der Einführung gibt es mehrere Möglichkeiten. Entweder können zunächst die agilen Themen weitergetrieben werden, also Kundennähe, Team-Arbeit usw. oder der Fokus geht zunächst auf die organisatorischen Themen und damit auf die Verfeinerung von Kanban mit den eingangs beschriebenen Anforderungen. Dies wird im nächsten Abschnitt dargestellt.

Ein umfassendes Konzept

Kanban kann, wie aus den bisher beschriebenen Zusammenhängen ersichtlich ist, also nicht ein eigenes Konzept sein. Viel mehr wird Kanban zu einem Bestandteil eines umfassenderen Konzeptes, in dem es insbesondere um die folgenden Themen gehen sollte:

Kosten-Management

Verzögerungskosten (COD) ist eine Kennzahl die im traditionellen Projektmanagement so nicht

„Aufgaben welche für die Organisation weniger wichtig sind als andere, werden hinten angestellt. Im Gegenzug dazu werden wichtigen Aufgaben in bisher ungeahntem Tempo fertiggestellt.“



auftaucht. Interessant ist es, folgende traditionellen Kennzahlen hinsichtlich ihrer wirtschaftlichen Auswirkungen zu hinterfragen: Einhaltung des Zeitplans, Einhaltung des Budgets, Durchlaufzeit in der Entwicklung, Qualität, usw.

Kürzere Durchlaufzeiten oder höhere Qualität bringen dem Unternehmen zunächst einmal nichts, denn das Kern-Interesse eines Unternehmens ist es, den Gewinn zu maximieren. Doch inwiefern wirken sich Durchlaufzeit oder Planeinhaltung auf diesen Gewinn aus? Qualitativ können diese Auswirkungen bei vielen Kennzahlen noch erkannt werden. Um im Projekt Entscheidungen zu treffen, müssen die Auswirkungen jedoch quantitativ benannt werden können. Man könnte fast annehmen, dass das klassische Projektmanagement sich eine Scheinwelt geschaffen hat, in der Dinge gemessen werden, die einfach zu messen sind, die jedoch keine betriebswirtschaftliche Aussagekraft haben.

Ein Projekt benötigt ein neues Gerüst an Kennzahlen die betriebswirtschaftlich fundiert sind. Nur so können schnell Entscheidungen getroffen werden. Die wichtigste Kennzahl ist dabei die bereits dargestellte COD. Hauptursache von COD in der Entwicklung sind unsichtbare Bestände: Warteschlangen zwischen den Projektschritten. Somit kann also alles was den Durchsatz verbessert, direkt in eine Steigerung des Projektgewinnes umgerechnet werden.

Auch für das Management des Durchsatzes sind neue Kennzahlen notwendig. Zum Beispiel die Länge der Warteschlangen. Dies ist das wichtigste operative Steuerungsinstrument. Warteschlangen sind ein sehr früher, sehr präziser Indikator für Probleme im Projekt. Die Messung der Durchlaufzeit wird hingegen erst auffällig, wenn die Probleme kaum mehr in den Griff zu bekommen sind.

Neben dem Management der Warteschlangen, dem der nächste Abschnitt gewidmet ist, gibt es weitere relevante Kennzahlen, wie zum Beispiel:

WIP, Losgröße, Auslastung, sowie Trends in den genannten Metriken.

Nicht nur das Projektmanagement muss sich nun auf betriebswirtschaftlich relevante Kennzahlen ausrichten, auch technologische Entscheidungen müssen anhand von Aussagen zum Projektgewinn getroffen werden können. Welchen Gewinn bringt zum Beispiel: Eine Gewichtsreduzierung um 100g? Eine Erhöhung der MTBF um 1000h? Eine Mikrosekunde weniger Zykluszeit auf dem Mikrocontroller? Alles Themenbereiche die oft nach Gefühl beantwortet werden.

Sicherlich ist es nicht immer einfach zu den aufgeführten Metriken gute Messungen bzw. Berechnungen durchzuführen. Es sollte jedoch allen Verantwortlichen bewusst sein, dass eine zielgerichtete Optimierung von Projekt und Produkt

unter betriebswirtschaftlichen Gesichtspunkten nicht möglich ist ohne zumindest ungefähre Aussagen zu den aufgeführten Zusammenhängen zu treffen.

Abschließend zu der Kostenanalyse noch ein wichtiger Punkt: Agile Ansätze wie zum Beispiel Scrum versuchen im Prinzip den Wert des Produktes ständig zu erhöhen.

Durch die absolute Betrachtung von Wert und Kosten ist es schwierig den besten Zeitpunkt für das Entwicklungsende zu erkennen (bezogen auf Eigenentwicklungen, bei Auftragsentwicklungen ist ein flexibles Projektende in der Regel nicht möglich). Besser ist es, in Grenz-Wert und Grenz-Kosten zu denken. Sobald der Grenz-Wert unter die Grenz-Kosten für eine zusätzliche Funktion fällt, sollte das Produkt als fertiggestellt betrachtet werden. Gute Product Owner bei Scrum treffen diese Abwägung in der Praxis, sie vergleichen für jedes zusätzliche Feature (nicht für das Produkt) Kosten und Nutzen.

Warteschlangen-Management

Wie bereits beschrieben bieten Warteschlangen einen sehr frühen Indikator für Unregelmäßigkeiten und sind als Frühwarnsystem viel besser

„Man könnte fast annehmen, dass das klassische Projektmanagement sich eine Scheinwelt geschaffen hat, in der Dinge gemessen werden, die einfach zu messen sind, die jedoch keine betriebswirtschaftliche Aussagekraft haben.“



geeignet als zum Beispiel die verbreitete Messung von Durchlaufzeiten. Auch wenn Warteschlangen-Management in der Entwicklung weitgehend unbekannt sein mag, so kennt es doch jeder als ein bewährtes Konzept in Supermärkten. Mitarbeiter im Supermarkt machen keine Projektpläne um festzulegen wer wann an die Kasse darf, sie messen auch keine Durchlaufzeit beim Bezahlvorgang. Die Mitarbeiter im Supermarkt messen die Länge der Warteschlange und öffnen eine weitere Kasse, sobald die Schlange eine gewisse Länge erreicht hat. Manche Supermärkte delegieren das sogar an die Kunden, mit einem Schild: „Mehr als 5 Kunden vor Ihnen? Bitte hier klingeln!“. Eine sehr elegante Lösung übrigens: Warteschlangen-Management mit dezentraler Messung und Entscheidung.

Gemäß der Beschreibung im Kapitel über Telekommunikationsunternehmen, stellen die berechneten Warteschlangen einen Durchschnitt dar. In der Praxis gehört daher zum Warteschlangen-Management ebenso ein Konzept zum Umgang mit volllaufenden Warteschlangen. Eine gute Option, insbesondere bei Testlaboren oder ähnlichem, ist es, ab einer bestimmten Warteschlangenlänge die Aufträge an externe Dienstleister zu vergeben, analog zum Supermarkt, der eine weitere Kasse öffnet.

Nicht immer ist jedoch die Möglichkeit zur Kapazitätserweiterung gegeben, dann geht es um die Überlegung unwichtige Aufgaben aus der Warteschlange zu entfernen, um die Wartezeit für wichtigere Aufgaben wieder in den erwarteten Bereich zu bekommen. Alternativ zum „Wegwerfen“ von Aufgaben (was bei internen Projekten einfacher ist als bei Kundenentwicklungen) kann die Priorisierung der Warteschlange verändert werden. Das Umsortieren einer Warteschlange stellt natürlich einen gewissen Aufwand dar und sollte durch bestimmte Regelwerke unterstützt werden.

Beim Umsortieren von Aufgaben in einer Warteschlange kommen auch wieder alte Bekannte zum Tragen: Cost of Delay (COD). Haben alle Aufgaben in der Schlange dieselben COD, muss zur Gewinnmaximierung die kürzeste Aufgabe zuerst bearbeitet werden. Sind hingegen die Aufgaben in etwa gleich lang und die COD unterschiedlich, muss die Aufgabe mit den höchsten COD nach vorne. In der Praxis gibt es meist nur Mischungen zwischen diesen beiden Fällen. Hier muss aus COD und Aufgabengröße eine Bewertung („Score“) berechnet werden um die Prioritäten in der Warteschlange festzulegen.

Warteschlangen-Management ist also eine Disziplin, die dem traditionellen Projektmanagement fehlt. Basis für das Warteschlangen-Management ist die Visualisierung und fortlaufende Messung der Warteschlangen. Agile Ansätze mit ihren Backlog-Konzepten sind dafür eine gute Lösungsoption.

Multiprojekt-Management

Scrum geht davon aus, dass es keine Multiprojekt-Umgebung gibt. Kanban hingegen ist als Werkzeug flexibel genug um verschiedene parallele Projekte darzustellen, liefert aber kein Konzept wie eine Multiprojektumgebung verwaltet werden soll. Hier muss also außerhalb der Agilität nach Lösungen gesucht werden.

Eine effektive Lösung für ein Multiprojekt-Management bietet das Critical-Chain Projektmanagement (CCPM), welches ebenso wie das DBR-Pullsystem aus der Engpassstheorie von Eliyahu Goldratt stammt. CCPM basiert zum einen auf dem Kern der Engpassstheorie, nämlich dem Schutz des Engpasses, und zum anderen auf einem expliziten Puffermanagement, im Gegensatz zu den impliziten Puffern in den klassischen Projektplänen.

Der Schutz des Engpasses ergibt sich aus der These, dass alles was der Engpass in einer Zeiteinheit nicht leisten kann, nie wieder aufgeholt

„Kanban hingegen ist als Werkzeug flexibel genug um verschiedene parallele Projekte darzustellen, liefert aber kein Konzept wie eine Multiprojektumgebung verwaltet werden soll. Hier muss also außerhalb der Agilität nach Lösungen gesucht werden.“



werden kann. Der Engpass, das langsamste Glied in der Kette, muss daher von unnötiger Arbeit befreit werden und vor der Organisation geschützt werden um möglichst effektiv arbeiten zu können. Während der Engpass in der DBR-Steuerung eine Maschine oder ein Prozessschritt ist, stellt beim Projektmanagement die kritische Kette den Engpass dar. Alles was auf der kritischen Kette verloren geht, kann nicht wieder aufgeholt werden. Hinweis: Die kritische Kette ist der kritische Pfad unter Berücksichtigung der verfügbaren Ressourcen. Der kritische Pfad bezieht sich nur auf die Abhängigkeiten der Arbeitspakete ohne Betrachtung der Ressourcen-Lage.

Ebenso wie der Engpass bei DBR als Instrument zur Steuerung eingesetzt wird, werden bei CCPM die Projekte in einer Multiprojektumgebung so angeordnet, dass die Engpass-Ressource immer nur ein Arbeitspaket eines Projektes zu einem Zeitpunkt bearbeiten muss. Auch bei CCPM ist also die Vermeidung von Multitasking einer der Schlüssel zum Erfolg. Alle anderen Arbeitspakete werden nach dem FIFO-Prinzip (First In First Out) vor und nach dem Engpass angeordnet. Ebenso wie Scrum und Kanban setzt CCPM also nicht auf Zeitpläne sondern auf die Sequenz der Tätigkeiten, einen Fluss. Natürlich müssen auch mit den Fluss-Methoden bestimmte Termine eingehalten werden können, es geht nur darum nicht jedem Arbeitspaket einen Zeitraum zuzuordnen. Die Reihenfolge wird über die Priorisierung in den Backlogs gesteuert.

Kanban und CCPM passen also sehr gut zusammen: Sie verzichten auf das Management von Timelines, was in der Praxis bei allem Aufwand ohnehin nur selten funktioniert.

Das zweite wichtige CCPM-Konzept ist das Puffermanagement. Begründet durch die meisten Unternehmenskulturen, fügen Mitarbeiter bei Schätzungen einen Puffer ein um ihre Zusagen halten zu können (die Schwierigkeit dabei wurde

eben indirekt erwähnt: In den meisten Kulturen gelten Schätzungen nicht als Schätzungen sondern als Zusagen). Das Problem bei diesen Puffern ist, dass deren Größe der Projektleitung nicht bekannt ist und sie dadurch die Puffer auch nicht verwalten kann: Der Projektpuffer verteilt sich auf viele kleine unsichtbare Puffer, die meistens auch von selbst wieder verschwinden. Hintergrund für letztgenannten Effekt sind psychologische Ursachen. Den Effekt der verschwindenden Puffer dürfte jeder Leser aus eigener Erfahrung kennen.

Die Idee von CCPM ist nun, Schätzungen ohne Puffer zu erheben und den Puffer als großen Block dem Projektplan hintanzustellen. Jedes Arbeitspaket darf nun (kulturell ungestraft) über die Schätzung hinaus andauern und sich sozusagen ein kleines Stück vom großen Projektpuffer nehmen. Der Projektplan schiebt sich dadurch Stück für Stück in den Puffer hinein. Das Projekt-

management muss nun sicherstellen, dass der Fortschritt auf der kritischen Kette stets größer ist als der aktuelle Pufferverbrauch. Dann kommt das Projekt sicher ins Ziel.

Die geplante Gesamtlaufzeit bleibt bei der CCPM-Planung dabei erst einmal gleich, in der Praxis ergeben sich jedoch Laufzeitverkürzungen im zweistelligen Prozentbereich durch das optimierte Puffermanagement: Der nun sichtbare

Puffer wird in der Regel nicht voll ausgeschöpft. Damit einher geht die Änderung der Kultur, d.h. jedes Paket kann in Ruhe fertiggestellt werden kann und Schätzungen bleiben wirklich Schätzungen. Dieses Vorgehen deckt sich mit den agilen Ansätzen und steigert spürbar die Qualität im Entwicklungsprojekt.

Zusammenfassung

Kanban ist ein eleganter und sanfter Ansatz, um Produkte mit Elektronik- und/oder Mechanik-Anteil effizienter zu entwickeln. Dabei wird die Mitarbeitermotivation erhöht ebenso wie die

„Die geplante Gesamtlaufzeit bleibt bei der CCPM-Planung dabei erst einmal gleich, in der Praxis ergeben sich jedoch Laufzeitverkürzungen im zweistelligen Prozentbereich durch das optimierte Puffermanagement.“



Qualität und der Projektgewinn. Ein blankes Kanban-System ist jedoch nicht ausreichend, es ist nicht agil. Um agil zu werden müssen weitere Aspekte in das Konzept aufgenommen werden.

Allerdings ist Agilität allein in komplexen Entwicklungsumfeldern nicht mehr ausreichend. Ökonomische Modelle müssen Entscheidungen im Projekt unterstützen, dazu müssen neue Kennzahlen eingeführt und bestehende Kennzahlen ignoriert werden.

Das Management von Backlogs ist zielführender als das Management von Projektplänen. Dies erfordert jedoch ein eigenes Konzept, welches sehr gut auf der Warteschlangentheorie aufgebaut werden kann. Letztgenannte liefert wertvolle Modelle und erklärt schlüssig bisher nicht wahrgenommene Zusammenhänge in Entwicklungsprojekten.

Für Multiprojektumgebungen bietet Critical-Chain Projektmanagement einen bewährten Ansatz. Durch eine hohe konzeptionelle Überdeckung mit Kanban bietet es sich an, CCPM und Kanban gemeinsam für das Management von Multiprojektumgebungen einzusetzen.

Über den Autor

Joachim Pfeffer ist Unternehmensberater im Bereich Lean und Agile Development. Nach über 10 Jahren in der Softwareentwicklung (klassisch, agil, embedded) und fünf Jahren Beratungspraxis in der Automobilindustrie (Functional Safety, SPICE, agil) sind seine Schwerpunkte agile System-Entwicklung, Lean Development und die Ökonomie von Entwicklungsprojekten. Als aktiver Berufspilot überträgt Joachim Pfeffer Team-Konzepte aus der Luftfahrt für Manager und Entwicklungsteams.



Kontakt

joachim.pfeffer@peppair.com

<http://joachim-pfeffer.com>

Literatur

Weiterführende Literatur für die in diesem Artikel vorgestellten Konzepte:

- Eliyahu Goldratt, „Das Ziel“, 2002
- David Anderson, „Kanban: Evolutionäres Change Management für IT-Organisationen“, 2011
- Donald Reinertsen, „The Principles of Product Development Flow: Second Generation Lean Product Development“, 2009
- Uwe Techt und Holger Lörz, „Critical Chain“, 2014
- Scrum Guide in der aktuellen Fassung, abrufen unter <http://www.scrumguides.org/>