

Henning Wolf  
Rini van Solingen  
Eelco Rustenburg

# DIE KRAFT VON SCRUM

Inspiration zur revolutionärsten  
Projektmanagement-Methode



ADDISON-WESLEY

## Ist Scrum etwas für uns?

Am nächsten Morgen wache ich früh auf. Ich habe gut geschlafen, aber ich scheine eine leichte Erkältung von der Nässe zu haben.

Noch im Bett liegend überdenke ich das Gespräch von gestern Abend und fange an zu sehen, was ich tun muss. Stefan hat recht: Wir brauchen Feedback von LogiStrux. Und das nicht auf der Basis von Dokumenten und Interviews, sondern auf Basis unseres Produkts. Das Problem ist, dass wir immer noch nicht annähernd fertig sind. Wir hatten einige Probleme in den letzten Monaten, zum Beispiel mit dem Kombinieren verschiedener Datenbanken für verschiedene Plattformen. Selbst jetzt sind da noch einige Fehler enthalten. Wir müssen außerdem unbedingt einen Webzugang bereitstellen, aber wir sind bisher daran gescheitert, die Berichte und Graphen in allen verschiedenen Browsern korrekt darzustellen. Die Berichte sind ja in sich schon schwierig genug, und wenn es uns gelingt, einen Bericht in einem Browser ordentlich darzustellen, sieht er im nächsten wieder schrecklich aus.

Allerdings bin ich der Meinung, dass dies die letzten 20% der Arbeit sind. Die anderen 80% haben wir erledigt, und wir sollten in der Lage sein, diese kurzfristig zu einem Release zusammenzubauen. Dazu sollten die 80%, die wir schon haben, für LogiStrux auch schon nützlich sein. Es wäre eigentlich super, wenn wir ihnen schon mal etwas zeigen könnten, weil es uns vertrauenswürdiger macht, wenn LogiStrux sieht, dass wir schon viel fertiggestellt haben. Gestern hatten einige der LogiStrux-Vorstände nämlich durchaus Zweifel daran. Sie schienen mir letztlich zu glauben, aber ihr Marketingvorstand war nicht überzeugt.

Was wäre, wenn wir alle problematischen Features weglassen und den Rest an LogiStrux als eine Beta-Version ausliefern? Natürlich müssten wir klarmachen, dass es sich nicht um das finale Produkt handelt, aber erst mal wären sie zufrieden und hätten schon etwas zum Testen. Außerdem könnten wir schon jetzt Feedback von ihnen bekommen und nicht erst in drei Monaten. Das ist zwar nicht komplett nach diesem Scrum-Ansatz, von dem Stefan sprach, aber es wäre doch eine prima Idee!

Ich stehe auf und gehe zum Frühstück. Vielleicht treffe ich meinen neuen Freund dort wieder. Vielleicht kann ich schnell meine Idee mit ihm dort diskutieren. Tatsächlich treffe ich ihn im Frühstücksraum. Er winkt mir zu und fragt mich, ob ich mit ihm zusammen frühstücken möchten. Eine Einladung, die ich dankend annehme.

„Ich habe darüber nachgedacht“, erzähle ich ihm, „ob wir ein Release unseres Produkts für einen unserer Kunden mit eingeschränktem Funktionsumfang ausliefern. Ist das eine gute Möglichkeit, um Feedback zu erhalten? Ich würde ihnen natürlich sagen, dass es ein Beta-Release ist!“

Stefan denkt einen Moment darüber nach und sagt schließlich: „Hm, das könnte eine gute Idee sein. Wenn ich dich richtig verstehe, würdest du ihnen die Funktionalität liefern, die schon ausreichend gut ist, dass man sie ausliefern kann, und alle problematischen Funktionalitäten würdest du entfernen. Ich würde ihnen aber nicht sagen, dass es eine Beta-Version ist. Das würde nur für euch und euren Kunden aussagen, dass die Qualität noch nicht gut wäre. Ich würde es eher ein Release eines Produktinkrements oder eines ‚Feature Sets‘ nennen. Das kommuniziert klar, dass es nur ein Teil des endgültigen Produkts ist, aber dass dieser Teil fertig ist. Das gibt dem Team und dem Kunden den richtigen Eindruck vom Zustand des Produkts. Das Team fokussiert sich dann mehr auf das aktuelle Release. Der Kunde kann beurteilen, ob die am meisten Wert schaffenden Produktfeatures richtig funktionieren. Das ist anders als eine ‚Beta‘. Man gibt seinem Kunden ein Produkt, oder man lässt es. Man sollte aber nie etwas ausliefern, was man selber nicht für gut hält. Denn das würde

bedeuten, dass du bereit wärest, schlechte Qualität zu akzeptieren.“

Er fährt fort: „Gleichzeitig höre ich aus deiner Situation heraus, dass ihr aktuell eine komplette Umstellung auf Scrum nicht durchführen könnt. Ich würde sagen, dass

du es eurem Kunden einfach mal vorschlagen solltest. Dann findest du doch heraus, ob sie dafür offen sind. Aber Vorsicht: Treffe explizite Vereinbarungen, was sie mit eurer Software tun werden. Stelle sicher, dass sie sie auch ausprobieren. Schließlich bekommst du kein Feedback, wenn sie sie nicht verwenden. Kein Feedback bedeutet aber nicht, dass alles in Ordnung wäre!“

„Aber ich denke insgesamt, dass es eine gute Idee ist“, setzt Stefan fort. „Ein testbares Produkt mit einigem an Funktionalität ist immer besser als kein Produkt. Stelle aber sicher, dass dein Release immer gute Qualität hat. Verhandle nie über Qualität. Nicht arbeiten ist besser als schlecht arbeiten! Wenn der erste Eindruck schlecht ist, dann musst du später doppelt so hart arbeiten, um diesen Eindruck in der Zukunft wieder zu verwischen.“

„Insbesondere die Oberflächen der Anwendung müssen perfekt sein. Wenn ein Teil nicht funktioniert, und das sich aber klar ergibt, wenn man draufklickt, ist es o.k.. In Scrum verwenden wir eine ‚Definition of Done‘, um das sicherzustellen. Aber das werde ich dir später erklären. Was immer ihr tut, das Release an den Kunden muss euren Qualitätsvorstellungen entsprechen. Darauf solltet ihr streng achten“, schließt Stefan.

Das ist natürlich auch mein Ziel. Ich werde am Montag als Erstes mit Christian darüber sprechen, dass wir unsere beiden Problembereiche aus dem Produkt entfernen und einen kompletten internen Test dieser Version des Produkts durchführen. Wenn wir die Version stabil

“**Ein testbares Produkt mit einigem an Funktionalität ist immer besser als kein Produkt. Stelle aber sicher, dass dein Release immer gute Qualität hat. Verhandle nie über Qualität. Nicht arbeiten ist besser als schlecht arbeiten!**”

“Scrum enthält nicht die Rolle des Projektleiters. Das Team ist verantwortlich für das Ergebnis und organisiert sich selbst. Der klassische Projektleiter als der ‚Chef‘ des Teams ist bei Scrum nicht mehr nötig.”

haben, melde ich mich bei LogiStrux und schlage ihnen ein spezifisch für sie erstelltes Pre-Release vor. Vielleicht schaffen wir es innerhalb eines Monats. Mal sehen, was wir erreichen können.

„Ich freue mich schon darauf, nächste Woche weiter mit dir zu sprechen“, sage ich. „Ich würde dir auch gerne Christian vorstellen, einen meiner Projektleiter. Er ist ein wirklich guter Mann. Es könnte eine gute Idee sein, wenn er sich dein Scrum anschaut und danach sein LogiStrux-Projekt organisiert!“

„Gut, aber ich bin mir nicht so sicher, ob er so begeistert sein wird“, antwortet Stefan, „denn Scrum enthält nicht die Rolle des Projektleiters. Das Team ist verantwortlich für das Ergebnis und organisiert sich selbst. Der klassische Projektleiter als der ‚Chef‘ des Teams ist bei Scrum nicht mehr nötig. Das Team macht seine eigene Planung für jeden Sprint basierend auf den Prioritäten des Product Owners. Das Team teilt die Arbeit selbst untereinander auf und macht auch den Fortschritt transparent. Für deinen Projektleiter wird Scrum eine wirklich große Veränderung.“

Ich verstehe das überhaupt nicht. Software-Entwicklung ohne Projektleiter? Das ist unmöglich! „Aber dann bekommst du doch den Viele-Köche-verderben-den-Brei-Effekt!“, sage ich zu ihm. „Ich brauche ganz bestimmt jemanden, der der Chef ist und in der Lage, den Entwicklern Arbeit zuzuordnen.“

„Ich befürchte, da liegst du falsch“, antwortet Stefan. „Du brauchst nur ein Team, das Verantwortung übernimmt und alles Notwendige tut, um den vereinbarten Termin einzuhalten. Software-Entwickler sind intelligente Leute. Sie werden nur gestört von jemandem, der sich wie der Chef aufführt. Es ist eine bekannte Tatsache, dass Wissensarbeiter dann am besten arbeiten können, wenn man ihnen erlaubt, sich selbst zu organisieren und Verantwortung zu überneh-

men. Wissensarbeiter müssen moderiert und nicht kontrolliert werden. Gib ihnen ein klares Ziel und die Freiheit der Selbstorganisation, und du bekommst ein System, das wie geschmiert läuft.“

„Was du vor allem brauchst“, führt Stefan fort, „ist eine klare Vision, was getan werden soll.

Und diese Vision muss aufgeschrieben werden im Sprint

Backlog und Product Backlog. Darüber hinaus brauchst du einen festen Arbeitsrhythmus für das Team. Der Arbeitsprozess kann eine Menge für dich tun. Du brauchst jemanden, der das Projekt unterstützt und sicherstellt, dass der Prozess rund läuft. In Scrum nennen wir so jemanden einen ‚ScrumMaster‘. Der ScrumMaster ist verantwortlich dafür, dass der Prozess gut funktioniert und der Rhythmus eingehalten wird. Der ScrumMaster ist aber nicht der Chef! Es ist eher eine moderierende Rolle, ganz bestimmt kein Projektleiter. Der ScrumMaster ergreift die Initiative, wann immer dies notwendig ist, und hilft, Probleme zu beseitigen. Den klassischen Projektleiter, der den Leuten sagt, was sie wann zu tun haben, gibt es in Scrum nicht. Der ScrumMaster ist sehr wichtig, weil er der Schmierstoff in unserem System ist, der dafür sorgt, dass alles rund läuft.“

Stefan schließt ab: „In der Praxis erlebe ich allerdings schon recht häufig, dass bei großen Projekten Anbieter weiterhin jemanden in der Rolle Projektleiter beschäftigen. Selbst wenn sie mit Scrum arbeiten! Dieser Projektleiter geht mit den Erwartungen des Kunden um, bringt Product Owner des Kunden und des Anbieters zusammen und koordiniert und unterstützt einen Steuerungskreis. So ein Projektleiter ermöglicht die Einbettung des Projekts im Rahmen einer strategischen Ausrichtung. Er verwaltet und vertritt das Projekt nach außen und stellt sicher, dass das Team geschützt arbeiten kann mit Scrum. Innerhalb des Teams ist diese Person aber nicht mehr der

“ Du brauchst jemanden, der das Projekt unterstützt und sicherstellt, dass der Prozess rund läuft. In Scrum nennen wir so jemanden einen ‚ScrumMaster‘. Der ScrumMaster ist verantwortlich dafür, dass der Prozess gut funktioniert und der Rhythmus eingehalten wird. Der ScrumMaster ist aber nicht der Chef!“

“Scrum ist der einzige Projektmanagementansatz, der auf dieser Detailebene Entwicklern erlaubt, ihre Arbeit selbst zu schätzen und zu planen, und ihnen hilft, ihre Versprechen einzuhalten.”

klassische Projektleiter, aber für das Team trotzdem sehr wertvoll. Zumindest wenn es ein guter ist.“ Stefan lächelt.

„Ich freue mich schon, nächste Woche mit dir weiter zu sprechen“, antworte ich. „Gibt es etwas, was ich bis dahin tun sollte?“

„Vielleicht“, antwortet Stefan, „wir haben bisher noch nicht über die Umgebung gesprochen, in der bei euch die Entwicklung stattfindet.“

„Scrum ist ein Ansatz, der sich wirklich nur mit dem *Management* des Software-Entwicklungsprozesses beschäftigt. Scrum ist dabei der detaillierteste, professionellste und strengste Projektmanagementprozess, den ich kenne. Scrum ist der einzige Projektmanagementansatz, der auf dieser Detailebene Entwicklern erlaubt, ihre Arbeit selbst zu schätzen und zu planen, und ihnen hilft, ihre Versprechen einzuhalten. Natürlich ist es in der Praxis sehr hilfreich, wenn man auf guten Entwicklungstechniken aufbauen kann, wenn man mit Scrum arbeitet. Ein Beispiel dafür ist ein vollautomatisierter Build-Prozess. Also eine korrekte Version vom Quellcode zu kompilieren, integrieren, eine Installationsversion zu erstellen etc. sollte idealerweise vollständig automatisiert werden. Man möchte dabei ja keine Zeit verlieren oder dabei Fehler machen. Als Zweites muss nach dem Zusammenbauen der Version immer ein ausführlicher automatisierter Test laufen. Wenn das Produkt den Test besteht, hat man ja jedenfalls ein gewisses Maß an Qualität sichergestellt. Ganz nebenbei hat man dazu Feedback erhalten, ob noch alles läuft seit dem letzten Build. Eine gute automatisierte Test- und Build-Umgebung ist wirklich notwendig. Wenn ihr so etwas nicht sowieso schon etabliert habt, dann ist es definitiv etwas, woran ihr sofort arbeiten solltet.“

Stefan erklärt weiter: „Wenn du ein Produkt inkrementell alle zwei Wochen ausliefern möchtest, dann musst du alle zwei Wochen über-

prüfen, dass das ganze Produkt auch noch funktioniert und sich die Qualität nicht verschlechtert hat. Das bedeutet, dass du dein komplettes Produkt alle zwei Wochen auch komplett testen musst. Ich bin mir sicher, dass du spätestens jetzt einsiehst, dass man das nur mit automatisierten Tests gut hinbekommt, weil sonst die Testaufwände mit jedem neuen Release steigen, weil ja auch jedes neue Release mehr Funktionalität enthält. Eine Lösung könnten externe Tester sein, die man alle zwei Wochen ins Haus holt, aber ich meine nicht, dass das die beste Lösung ist.“ Er lacht leise. „Wenn du deine Tests automatisierst, bist du dir jederzeit der Qualität deines ganzen Produkts bewusst.“

Ich weiß nicht, woher Stefan seine Erfahrungen hat, aber er scheint ein Hellseher zu sein. In unserem letzten Release hatten wir viele Probleme in diesem Bereich. Der Entwickler, der sonst immer die letzten Dinge am Release gemacht hat, damit wir es auf DVD brennen konnten, war mit gebrochenem Bein zu Hause. Als wir den ersten Stapel an DVDs erhielten, gab es leider einige ernsthafte Probleme, auch Fehler im Installationsprogramm. Alle DVDs mussten weggeschmissen werden. Nach dieser Erfahrung haben wir unseren Build-Prozess um einiges verbessert. Wir sind noch nicht so weit, dass man nur einen Knopf drücken müsste, aber wir kommen dem schon ziemlich nah. Wir sollten es in ein paar Tagen hinbekommen. Das Aufsetzen automatisierter Tests wird schwieriger, aber ich weiß, dass einige der Tester da schon irgendwas angefangen haben. Vielleicht ist die Dringlichkeit des LogiStrux-Projekts eine gute Gelegenheit, das jetzt mal ernsthaft in Angriff zu nehmen.

„Die Testumgebung wird etwas dauern“, sage ich, „aber wir fangen damit an. Wenn wir so ein System etablieren, würde es doch auch Sinn machen, wenn wir beim Entwickeln auch konsequent Unit-Tests einsetzen. Wenn die Entwickler inhaltlich sowieso mit dem Testen beschäftigt sind, dann können sie doch diese Tests auch gleich mit erstellen. Diese Tests können wir dann gleich in die Testumgebung übernehmen, um unsere Qualität für jedes neue Build zu überprüfen.“

„Du lernst sehr schnell.“ Stefan lächelt. „Das nennen wir testgetriebene Entwicklung oder TDD für ‚Test-driven Development‘. Das ist



auch wieder so was, was eigentlich nicht Teil von Scrum ist, aber trotzdem sehr nützlich. Egal, ich muss jetzt wirklich los, mein Flug geht in anderthalb Stunden!“

Ich weiß aus Erfahrung, dass 90 Minuten nicht immer ausreichen, um zum Flieger zu kommen, und empfehle ihm, sich zu beeilen. Aber es ist Samstag morgen, da wird es wohl nicht so voll sein in Heathrow.

\*\*\*

Am nächsten Montag bin ich früh im Büro. Meinen Flieger habe ich diesmal gut bekommen und konnte fast den ganzen Flug schlafen. Ich sitze schon seit vor sieben im Büro. Die meisten Kollegen sind noch nicht da, es ist angenehm ruhig.

Der Einzige, der auch schon so früh da ist, ist Markus. Er ist einer unserer Tester und noch ziemlich jung. Vor drei Jahren hat er seinen Abschluss an der Universität gemacht, er ist hochmotiviert und technisch sehr versiert. Er ist sehr freundlich, und die meisten mögen ihn. In den letzten Monaten habe ich ihn ein paar Mal gebeten, dass er das Produkt Gästen zeigt oder neue Features unseren Verkäufern. Er ist geduldig und kann sowohl mit erfahrenen wie auch mit unerfahrenen Anwendern gut umgehen. Was auch sehr wichtig ist: Er kennt unser Produkt durch und durch und kann sehr gut über seine Verwendung nachdenken.

Ich habe schon mal beobachtet, dass selbst Entwickler zu Markus gehen, um ihn zu fragen, wie bestimmte Features genau funktionieren. Man würde ja erwarten, dass es die Entwickler besser wissen, aber es ist Markus, der alle Tipps und Tricks kennt und sich immer Zeit nimmt zu helfen. Dabei bringt er dadurch seine eigene Arbeit nicht in Gefahr. Er ist immer pünktlich fertig, und wenn etwas sich zu verspäten droht, weist er früh darauf hin. Seine persönlichen Planungs- und Kommunikationsfähigkeiten sind wirklich super.

„Hallo Chef, du bist aber früh hier heute!“ Markus steht in meinem Büro. „Das Wochenende in London genossen?“

„Ja, das war wirklich ein großartiger Urlaub!“, antworte ich. „Von Donnerstag auf Freitag habe ich gar nicht geschlafen. Direkt vom Flughafen zu LogiStrux, um von denen auseinandergenommen zu werden. Anschließend durchnässt im Regen und meinen Flieger verpasst. Ja, war wirklich super, das Wochenende. Du solltest es auch mal probieren irgendwann!“

„Ach, nein danke“, antwortet Markus. „Ich bleibe mit meinen Füßen lieber auf dem Boden. Ich bin kein Held im Flugzeug. Aber ich habe was anderes. Ich glaube, dass wir ein Problem haben. Ich habe mir Christians neuen Plan angeschaut, und wir haben demnach nur zwei Wochen zum Testen. Ich habe mit ihm darüber gesprochen, aber er sagt, dass mehr nicht drin ist. Du weißt, wie es letztes Mal lief. Wir brauchen mindestens sechs Wochen. Es ist ja nicht nur das Testen. Was wir finden, muss ja auch repariert werden, und anschließend müssen wir es wieder testen, um den Fehlerbericht schließen zu können. Zwei Wochen sind wirklich zu knapp, aber Christian sagt, dass es keine Alternative gibt. Er würde ja gerne, sagt er, aber es passt einfach nicht rein.“

„Ich spreche mit Christian darüber“, antworte ich. „Aber ich würde mit dir gerne noch über etwas anderes sprechen heute Nachmittag. Ich möchte, dass du eine Testumgebung aufsetzt, die wir automatisch nach jedem Build ausführen können.“

„Was meinst du genau?“, fragt er. „Einen automatisierten Regressions-test? Oder mehr?“

„Ich möchte, dass es eine Umgebung ist, die mehr enthält als Regressionstests“, antworte ich. „Es wird auch einen funktionalen Regressionstest darin geben, aber ich möchte auch, dass alle Entwickler ihre automatisierten Unit-Tests einbinden können. Das Ziel ist, dass wir uns einem ausführlichen Test nähern, den wir nach jedem neuen Build auf jeder neuen Version ausführen können. Das ist deutlich mehr Automatisierung, als wir heute haben. Wir müssen dahin kommen, dass wir nur noch auf einen Knopf drücken, und dann wird ein neues Release erstellt und komplett automatisch durchgetestet. Das erlaubt uns dann, die Qualität mit jedem Build zu messen.“

„Klingt super!“, antwortet Markus. „Ich habe mit so was selbst auch schon rumgespielt. Ich habe mein eigenes Test-Setup bereits automatisiert laufen. Das spart mir oft sehr viel Zeit. Ich lasse es immer laufen, wenn mich jemand bittet, ein neues Build zu testen. Wenn das Produkt dann diese Basistests schon nicht besteht, ist es für mich sinnlos, den Rest überhaupt weiter anzuschauen. Ich setze mich heute Nachmittag mit Susan zusammen, um zu besprechen, ob das für die Entwickler auch ein passender Weg ist.“

Susan ist eine unserer erfahrensten Entwicklerinnen und immer offen, neue Dinge auszuprobieren, allerdings nur, wenn es für sie auch pragmatisch und funktionierend aussieht. Susan scheint eine gute Wahl zu sein. Ich spreche noch etwas mehr mit Markus darüber, wie nützlich eine solche Testumgebung sein könnte. Ein Termin mit Markus heute Nachmittag ist gar nicht mehr nötig. Wir vereinbaren, dass Markus mir in ein paar Tagen berichtet, was möglich ist.

Ich öffne meine E-Mail. Glücklicherweise bin ich meine Inbox gestern Abend schon durchgegangen. Nichts ist schlimmer, als die Woche mit einer vollen Inbox ungelesener Mails zu beginnen. Es gibt nur eine neue Nachricht: Eine Nachricht aus Hamburg.

Stefan kommt am Dienstag nach München und bleibt bis Sonntagabend in der Stadt. Er schlägt vor, dass wir uns Donnerstagabend zum Essen treffen. Er schlägt auch vor, dass er uns am Freitag hier im Büro besuchen kommt. Ich antworte ihm und bestätige den Essenstermin am Donnerstag. Freitagnachmittag habe ich nur wenig Zeit, aber ich frage ihn trotzdem, ob er da kommen mag.

Ich denke, dass er sich mit Christian bekannt machen kann und die beiden vielleicht zusammen Ideen entwickeln, wie wir das Logi-Strux-Projekt verbessern können.

Ich schicke einen Termin für Freitagnachmittag an Christian und auch einen an Markus für eine Produktdemo unseres Besuchers aus dem Norden. Christian antwortet sofort: Er wird da sein. Markus bestätigt nach ein paar Minuten auch seine Teilnahme.

# Die Essenz

- ▷ Bei Scrum ist das Software-Entwicklungsteam vollständig verantwortlich für alle Ergebnisse und organisiert sich deshalb selbst. Das Team entscheidet über seinen Prozess und die verwendeten Werkzeuge. Was das Team baut, bestimmt der Product Owner, wie es das aber erledigt, entscheidet das Team selbst.
- ▷ Der ScrumMaster ist verantwortlich, dass der Scrum-Prozess funktioniert und rund läuft und einem Rhythmus folgt. Der ScrumMaster ist aber nicht der Chef.
- ▷ Bei Scrum ist der traditionelle Projektleiter als Chef nicht mehr nötig. Das Team organisiert sich während des Projekts selbst. Der ScrumMaster ist nur verantwortlich, dass der Prozess effizient und effektiv läuft.
- ▷ Das Team plant jeden Sprint auf Basis der Prioritäten des Product Owners selbst. Das Team teilt die Arbeit selbst auf und überwacht den Fortschritt auf nachvollziehbare und transparente Art und Weise.
- ▷ Scrum ist ein detaillierter und rigoroser Projektmanagementprozess. Scrum ist der einzige Ansatz für Projektmanagement, der in diesem Detailgrad den Entwicklern erlaubt, ihre Arbeiten selbst zu planen und abzuschätzen, und ihnen dabei hilft, ihre Versprechungen einzuhalten.
- ▷ Der Arbeitsprozess ist bei Scrum sehr diszipliniert. Die Selbstorganisation des Teams ist dabei in der Methode selbst angelegt.
- ▷ Die Rolle des Projektleiters bleibt in einer Außensicht hilfreich. Allerdings gestaltet sie sich dann ein wenig anders. So ein Projektleiter fokussiert auf die Umgebung des Projektes und koordiniert externe Dinge wie Berichte, Steuerungsgruppen, Anfragen von außen und Eskalationen. Innerhalb des Teams ist diese Rolle aber nicht mehr die des klassischen, für das Team Entscheidungen treffenden Projektleiters, aber für den Kunden und das Entwicklungsteam kann das trotzdem wertvoll sein.