Achtung: Satire!

How to kill agile: Die Anleitung für Saboteure

Clemens Mucker

Agile Vorgehensmodelle sind in aller Munde – aber nicht jedermanns Sache. Wie Sie als Gegner von agilen Projekten ein solches gezielt sabotieren können, lesen Sie in dieser satirischen Anleitung. Oder Sie lernen fürs nächste Mal, wie ein solches Projekt erfolgreich und agil abgewickelt werden kann.

In der Softwareentwicklung gibt es seit geraumer Zeit (wieder) eine neue Wunderwaffe: agiles Vorgehen. Diese beinhaltet Schlagwörter wie Scrum, Extreme Programming (XP), Agilität und Flexibilität, schnelle Ergebnisse und wenig Aufwand. Doch wie jedes Werkzeug müssen auch agile Vorgehen richtig angewendet werden. Sie bergen großes Potenzial, aber auch viele Risiken. Die optimale Grundlage also, wenn Sie ein agiles Projekt scheitern lassen wollen! Durch die schrittweise Befolgung der Top 12 meiner Hitliste erzielen Sie größte Erfolgschancen für Ihre Sabotage.

Top 1: Reden ist Silber, Schweigen ist Gold

Gerade in agilen Projekten ist Kommunikation der Dreh- und Angelpunkt. Nicht umsonst wird die Face-to-Face-Kommunikation als die effektivste beworben. Agile Teams sitzen möglichst gemeinsam in einem Raum – lokalisiert und zentriert. Wenn Sie ein agiles Projekt sabotieren wollen, sorgen Sie gleich zu Anfang dafür, dass die Teammitglieder möglichst weit auseinander sitzen – verschiedene Räume sind gut, verschiedene Länder noch besser, unterschiedliche Zeitzonen der absolute Traum für jeden Saboteur! Während die einen Teammitarbeiter bereits schlafen, sitzen die anderen gerade beim täglichen Stand-up.

Befürworten Sie daher möglichst interkontinentales Outsourcing, denn sprachliche und kulturelle Barrieren helfen Ihnen ebenfalls, das Projekt zum Scheitern zu bringen.

Top 2: „TEAM“ = Toll, Ein Anderer Macht´s

Durch unterbundene Kommunikation kann die „Toll, ein anderer macht´s“-Teammentalität perfekt gedeihen. Richtige Teamarbeit soll durch die agilen Ansätze gefördert werden und ist Grundbaustein eines jeden Projekts. Wenn niemand weiß, was der andere macht, sind gute Voraussetzungen zum Scheitern gegeben. Wenn sich Ihre Teammitglieder nicht abstimmen, wer sich eines Problems annimmt, und sich jeder darauf verlässt, dass es schon irgendjemand übernehmen wird, ist Ihnen der Erfolg, das Projekt zum Scheitern zu bringen, so gut wie sicher. Bei besonders interessanten Problemen kann es vorkommen, dass sich alle Ihre Mitarbeiter darauf gleichzeitig stürzen. Der Umstand braucht Ihnen keine Sorgen zu bereiten, denn so wird wiederum wertvolle Projektzeit und Budget vergeudet, welches mit hoher Sicherheit gegen Ende des Projekts fehlen wird.

Achten Sie auch auf die außerberufliche Kommunikation. Gemeinsame Aktivitäten nach der Arbeit sind zu vermeiden. Die Mitarbeiter könnten sich anfreunden und miteinander sprechen, sich vielleicht sogar verbünden.

Top 3: Auswahl der Mitarbeiter

Ein weiteres probates Mittel liegt in der richtigen Auswahl der Mitarbeiter. Suchen Sie lieber nicht den motivierten, verantwortungsbewussten Teamplayer, dieser macht Ihnen die Arbeit nur unnötig schwer. Suchen Sie lieber Einzelkämpfer, Platzhirsche, Alpha-Männchen, absolute Herrscher. Abgesehen davon, dass sich Ihre Mitarbeiter nicht mehr freuen, zur Arbeit zu kommen, werden sie auch alles versuchen, die anderen auszustechen und gehen über virtuelle Leichen. Anstatt gemeinsam am Projektziel und -erfolg zu arbeiten, zermürben sie sich in nutzlosen Kleinkriegen untereinander.

Das wiederum können Sie fördern, indem Sie Wettbewerbe ausrufen. Aber bitte möglichst sinnlose! Beispiele hierfür sind: Wer schreibt mehr Code(s) am Tag? Wer schreibt mehr Testfälle? (Wer sie später durchführt …, danach haben Sie ja – noch – nicht gefragt). Ebenfalls ein guter Wettbewerb: Wer findet mehr kritische und damit auslieferungsverhindernde Fehler? So wird plötzlich jeder Fehler kritisch – sehr zur Freude der Entwickler und Projektleitung. Es werden endlose Diskussionen entstehen, ob denn die Einstufung korrekt ist.

Top 4: Unfreiwillige Freiwillige

Agile Methoden bauen auf Selbstverantwortung der Beteiligten. Versuchen Sie daher, Ihre Mitarbeiter möglichst durch Zwangsrekrutierung zu gewinnen. Besonders geeignet sind

Exkurs: Das agile Manifest

Die ersten Ansätze zur agilen Softwareentwicklung stammen aus den Neunzigern und firmierten unter dem Begriff „leichtgewichtig“. Der Begriff der agilen Softwareentwicklung wurde 2011 bei einer Konferenz, in der auch das Agile Manifest (siehe `http://agilemanifesto.org/iso/de/`) formuliert und unterzeichnet wurde, geprägt. Dieses Manifest besagt Folgendes:

„Wir erschließen bessere Wege, Software zu entwickeln, indem wir es selbst tun und anderen dabei helfen. Durch diese Tätigkeit haben wir folgende Werte zu schätzen gelernt:

Individuen und Interaktionen	*mehr als*	*Prozesse und Werkzeuge*
Funktionierende Software		*umfassende Dokumentation*
Zusammenarbeit mit dem Kunden		*Vertragsverhandlung*
Reagieren auf Veränderung		*das Befolgen eines Plans*

Das heißt, obwohl wir die Werte auf der rechten Seite wichtig finden, schätzen wir die Werte auf der linken Seite höher ein.“

Vor allem der letzte Punkt wird oft vergessen. Perfekt für ein Sabotage-Vorhaben.

Exkurs: Phasenmodell der Teambildung nach Tuckmann

Nach dem Phasenmodell der Teambildung von Bruce Tuckmann (`http://de.wikipedia.org/wiki/Teambildung#Phasenmodell_nach_Tuckman`) durchläuft jedes Team der Reihe nach folgende Stufen: Forming – Storming – Norming – Performing – Adjourning. In der Leistungsphase „Performing" arbeitet das Team geschlossen am gemeinsamen Ziel. Die Teammitglieder kooperieren und helfen sich gegenseitig, es herrscht eine harmonische Atmosphäre verbunden mit Anerkennung und Wertschätzung.

Wird ein neues Teammitglied aufgenommen oder ändert sich die Aufgabenstellung, beginnt der Durchlauf des Phasenmodells von vorne. Diese Durchläufe erfolgen meist schneller, jedoch erneut von Beginn an.

Um Ihr Projekt zu sabotieren, erreicht Ihr Team durch regelmäßige Änderungen der Zusammensetzung idealerweise nie die Phase „Performing", sondern maximal die Konfliktphase „Storming", in der Machtkämpfe und Spannungen vorherrschen.

dabei Menschen, die Wert auf akademische Titel legen und selbst bereits als Projektmanager gearbeitet haben. Sie werden sich meistens mit Freuden in die flache Hierarchie einfügen und voller Motivation im Team arbeiten.

Unfreiwillige zu verpflichten, gilt natürlich auch für Kunden. Da Sie auf die Mitarbeit der Kunden angewiesen sind – denken wir nur mal an den Product Owner –, zwingen Sie ihn einfach die agilen Methoden mitzumachen. Da Kunden meistens möglichst wenig Aufwand auf ihrer Seite wollen, sind sie sicher erfreut über den Vorschlag, einen ihrer Mitarbeiter Vollzeit als Product Owner für die gesamte Projektlaufzeit anzustellen. Wenn der Mitarbeiter noch dazu notorisch überlastet und daher nie für Rückfragen erreichbar ist, haben Sie Ihren idealen Product Owner gefunden.

Top 5: Teilzeit-Mitarbeiter

Ebenfalls hat sich die Methode bewährt, Mitarbeiter nicht Vollzeit im Projekt zu beschäftigen, sondern nur sporadisch. Idealerweise nehmen diese nicht an Planungsmeetings teil und opfern statt dessen regelmäßig einige Stunden ihrer Teilzeit, um auf den neuesten Projektstand zu kommen. Kaum könnten sie produktiv tätig sein, ist leider ihr Stundenkontingent aufgebraucht.

Mitarbeiter, die etwa nur zu 10 Prozent an Ihrem Projekt mitwirken, sollten Sie so verplanen, dass diese Prozent möglichst nicht am Block verwendet werden können. So verbringen diese Mitarbeiter den meisten Teil ihrer Zeit damit, sich auf den neuesten Stand zu bringen, ohne selbst produktive Beiträge leisten zu können.

Top 6: ÜberROLLEN

Wie zuvor erwähnt, sollte der Product Owner von Ihrem Kunden eingestellt werden. Am besten ein Mitarbeiter, der keine Erfahrung mit der Rolle besitzt und außerdem kundenintern viel um die Ohren hat. Sie werden ihn selten erreichen und der Product Backlog wird nicht gewartet, wetten? Wenn er dazu noch entscheidungsschwach ist und die Anforderungen nicht priorisieren kann, dann haben Sie in diesem Punkt bereits gewonnen.

Dem Product Owner stellen Sie einen Scrum Master aus Ihren Reihen zu Seite. Ebenfalls sehr gut geeignet ist jemand, der keine Erfahrung mit Scrum, sondern nur mit klassischem Projektmanagement hat. Besonders vorteilhaft ist es, wenn der Scrum Master weniger Ahnung von Scrum hat als der Rest des Teams. Wenn er dazu noch etwas einsiedlerisch veranlagt ist und sich gerne in technischen Details verliert, kann Ihnen das nur recht sein für Ihren Sabotageakt. Ändern Sie nach Möglichkeit oft die Teammitglieder. So erreicht das Team nie die volle Leistung, da es nie in die Performing-Phase kommt.

Top 7: „Wir sind fertig!"

Mit Freude wird dieser Ruf erwartet. Die ganzen Mühen, der ganze Schweiß in vollgepferchten Räumen … Endlich sind die Stories fertig. Sie dürfen sich freuen. Idealerweise gibt es nämlich keine verbindliche Definition of Done. Sie ist nur sehr vage formuliert, es ist nicht klar, wer „Done" sagen darf. Da der Product Owner letztlich die Verantwortung trägt und mit Hilfe der Tester die Abnahme durchführt, empfiehlt es sich, wenn Entwickler die Entscheidung treffen. Es gibt nämlich einige Punkte, die nur zu gerne vergessen werden – der Test, die Dokumentationen oder der Endanwender.

Top 8: Agile Methoden brauchen keine Dokumente

Das weit verbreitete Vorurteil „Agile Methoden brauchen keine Dokumentation" können Sie sich nicht nur bei der Definition of Done zu Nutze machen: Product und Sprint Backlog müssen nicht unbedingt gewartet werden – zumindest nicht, wenn Sie das Projekt scheitern lassen wollen. Es empfiehlt sich, die – natürlich nicht gewarteten – Backlogs zur Sicherheit möglichst versteckt (quasi wartfrei) anzubringen, oder Sie führen Backlogs gar nicht erst ein ... Aber was, wenn doch? Keine Panik! Sie haben noch die Möglichkeit, im Product Backlog frei nach Michael Ende never ending stories anzulegen, nicht zu priorisieren, oder die Stories nicht auf umsetzbare Einheiten herunter zu brechen. Denken Sie daran – wenn Sie alles richtig gemacht haben, ist der Product Owner nicht erreichbar und für Ihre Unterstützung dankbar.

Beim Sprint Backlog haben Sie ebenfalls viele Einflussmöglichkeiten. Lassen Sie den Sprint Backlog nicht durch das Team befüllen. Ändern Sie ihn regelmäßig und bringen Sie ihn nicht auf den neuesten Stand. Auch hier wird der Scrum Master für Ihre Unterstützung dankbar sein. Und Burndown Charts … Zeichnen Sie einfach bunte Linien und vertrauen Sie darauf, dass sowieso niemand versteht, was diese bedeuten. Andere Formen der Dokumentation wie zum Beispiel Testfälle und Mängel sollten Sie möglichst vermeiden. Sollen sich die Teammitarbeiter doch merken, was sie getan haben.

Ebenfalls empfiehlt es sich, Änderungen bei User Stories nicht nachzutragen, so bleibt der aktuellste Stand gut versteckt. Änderungswünsche, die zwischen Tür und Angel und nur zwischen zwei Leuten ausgemacht wurden, haben zudem den Vorteil, dass sie die Projektkosten ungemein in die Höhe treiben können.

Top 9: Flexible Änderungen

Agilität bedeutet Flexibilität und diese können Sie am besten zeigen, indem Sie Änderungen an Anforderungen oder User Stories bis zuletzt einbringen und annehmen (lassen). Idealerweise befindet sich die User Story bereits im Test, bevor sie erneut geändert wird, am besten kurz vor einem Produktiv-Deployment. Um das Projekt zu sabotieren, können Sie aber nicht nur bis zuletzt Funktionalität hinzufügen. Oft reicht es, wenn aus User Stories Funktionalitäten entfernt werden. Sehr gerne wird nämlich im weiteren Verlauf des Projekts der Bezug auf diese Funktionen vergessen – vor allem, wenn Sie noch dazu verhindern können, dass für diese Auslagerungen eine neue Anforderung oder User Story angelegt wird.

Kurz vor Produktivsetzungen entfernte oder unterbundene Funktionalitäten sorgen auch für einen schönen Wirbel, wenn die Endanwender durch das Ausblenden von Funktionalitäten plötzlich ihre Arbeit nicht mehr machen können. In beiden Fällen sollten Sie allerdings ein Auge auf die Tester haben, denn diese haben oft die unangenehme Angewohnheit, Ihre Sabotage wiederum zu sabotieren und darauf hinzuweisen, dass hier relevante Funktionalitäten fehlen.

Top 10: Viele Werkzeuge – viel Verwirrung

Wenn Sie auf die Dokumentation vollständig verzichten, benötigen Sie weniger Werkzeuge. Falsch! Schaffen Sie möglichst viele teure und komplizierte Werkzeuge an. Abgesehen davon, dass diese Anschaffungen natürlich das Budget belasten (Anschaffung, Schulung und Wartung), werden es Ihnen auch die Mitarbeiter danken! Verzichten Sie dabei auf einfache Mittel wie Pinnwände für Backlogs, Karteikarten für die Stories, setzen Sie auf die neueste Technologie. Je weniger diese verstanden werden und je benutzerunfreundlich diese sind, desto besser für Sie. Verwenden Sie möglichst viele verschiedene Tools, die nicht miteinander kommunizieren können.

Nur in einem Punkt sollten Sie auf keinen Fall ein Werkzeug besorgen: für die Automatisierung! Dann ist Ihnen die Sabotage definitiv gelungen. Leider setzt sich die Erkenntnis immer mehr durch, dass gerade bei agilen Projekten die Automatisierung von Testfällen essenziell ist. Aber ohne Tool kann nicht automatisiert werden, und die Tools für die Automatisierung sind viel zu teuer. Das Team kann auch so testen und ein Regressionstest wird nicht benötigt.

Top 11: Management-Involvement

Vor allem bei den Kosten für die Automatisierung wird Ihnen das Management freudig zustimmen, ist es doch über jeden gesparten Euro froh! Überhaupt sollten Sie das Management intensiv mit an Bord holen. Sie sollen sich möglichst oft einmischen zum Beispiel den Sprint Backlog befüllen. „Da geht noch mehr“, sollte dabei die Devise sein.

Das Management eignet sich auch besonders gut, um eventuell vorhandenen Zusammenhalt im Team zu zersplittern. Je weniger die hinzugezogenen Manager dabei von agilen Methoden verstehen, desto besser. Dann gilt die erste Planung auch schon mal in Stein gemeißelt. Außerdem hinterfragt niemand, ob die Planung vom gesamten Team stammt oder ob die Planung nicht am grünen Tisch von vielleicht sogar Projektfremden gemacht wurde! Versuchen Sie, dass die Schuld für eine fehlerhafte Planung nicht bei Ihnen gesucht wird. Das können Sie notfalls erreichen, indem Sie zwar dem Team die Planung überlassen, dabei aber Aufwände für Test und Dokumentation nicht berücksichtigen (demnach nur Entwickler einladen, zu schätzen) beziehungsweise die Planung erfolgt durch das Management und das Team darf sich dazu nur mehr committen. In diesem Fall ist natürlich das Team schuld, wenn der vereinbarte Umfang nicht umgesetzt werden kann.

Top 12: Aber der Prozess sagt …

Halten Sie sich sklavisch an den vorgegebenen Prozess! Agile Methoden sind gut und schön, aber muss daran so viel agil sein? Passen Sie die Methode nicht an Ihr Umfeld und Ihr Projekt an, passen Sie das Projekt an die Methode an. Versuchen Sie sich in eine Hose zu quetschen, die drei Größen zu klein ist, anstatt sich eine passende Hose zu besorgen. Selbstverständlich sind die in den Büchern vorgeschriebenen Prozesse und Methoden heilig und dürfen keinesfalls abgeändert werden, schließlich ist der Autor ein Experte. Es empfiehlt sich, einige Bücher zum Thema „Agile“ im Vorfeld zu lesen und das passendste auszusuchen. Erklären Sie dieses Buch zur Heiligen Schrift, jede Kritik daran wird unterbunden, jeder Verbesserungsvorschlag abgeschmettert und jede Anpassung an Ihr Projekt abgewiesen.

Bestehen Sie auch regelmäßig auf Prozessverbesserungen. Meistens werden die Prozesse dahin gehend verschlimmbessert, dass sie noch genauer definiert werden. Was in ruhigeren Projektphasen Sicherheit und Durchblick gibt, ist für hektische Phasen oft zu langwierig. Daher werden in solchen stressigen Projektphasen Prozesse meist nicht eingehalten, keine Änderungen dokumentiert und viel „unter der Hand“ erledigt. Dabei handelt es sich um Versäumnisse, die sich meist zu einem späteren Zeitpunkt unvorteilhaft auswirken, zum Beispiel wenn die daran beteiligten Teammitglieder ausgetauscht werden und niemand mehr von den Änderungen in Kenntnis gesetzt wird.

Fazit

Wenn Sie diese Ratschläge beherzigen, können Sie jedes agile Projekt garantiert scheitern lassen. Handelt es sich dabei sogar um das Pilotprojekt für die unternehmensweite Einführung von agilen Vorgehen, haben Sie vermutlich Ihre gesamte Organisation gerettet.

Und wenn Sie Ihr Projekt eigentlich gar nicht sabotieren wollten, wissen Sie jetzt wenigstens, warum es dennoch gescheitert ist. Denn die besten Saboteure sind jene, die nicht einmal wissen, dass Sie das Projekt sabotieren.

Clemens Mucker kam im Jahr 2000 über Umwege zum Softwaretest. Nach Projekten in Deutschland und der Schweiz ist er seit 2009 als Senior Test Consultant für ANECON tätig. Zu seinen Aufgaben zählen unter anderem Coaching und Testmanagement mit Schwerpunkt auf der Einführung von Softwaretests in Unternehmen und dem Aufbau von Testteams.
E-Mail: clemens.mucker@anecon.com