



DIGITAL TRANSFORMS PHYSICAL

Skalieren agiler Entwicklung: neun Fehler, die es zu vermeiden gilt

Lektionen zur agilen Transformation
von weltweit führenden
Fertigungsunternehmen



Einführung

Agile Entwicklung ist in der Welt der Softwareentwicklung der methodische De-facto-Standard.

Laut einer **Umfrage von McKinsey** aus dem Jahr 2021 erzielten Organisationen durch eine hochgradig erfolgreiche agile Transformation in der Regel eine Effizienzsteigerung von 30 %, arbeiteten fünf- bis zehnmal schneller und brachten Innovationen in deutlich kürzerer Zeit auf den Weg. Darüber hinaus war es 3-mal wahrscheinlicher, dass diese Organisationen mit ihrer Leistung ins Spitzenquartil aufschlossen.

Diese Statistik ist so überzeugend, dass Organisationen agile Projektmanagementtechniken zunehmend nicht mehr nur in der Softwareentwicklung, sondern auch darüber hinaus einsetzen, beispielsweise im Marketing, im operativen Betrieb, in der Hardware und – Schwerpunkt dieses E-Books – in der Produktentwicklung.

Produktentwicklungsteams werfen ihre kollektive Kreativität in die Waagschale, um bahnbrechende Produkte zu schaffen. Ob bei Flugzeugen, Automobilen, Staubsaugern, Herzschrittmachern, Insulinpumpen oder Roboterarmen – Produktentwicklungsteams haben viel gemeinsam.

Was bedeutet agil?

- **Eine Projektmanagementstrategie, die den Schwerpunkt auf Flexibilität, Zusammenarbeit und Anpassungsfähigkeit legt**
- **Eine Haltung, die auf kontinuierliche Verbesserungen ausgerichtet ist**
- **Eine Reihe von Frameworks, die agile Ideale wie Scrum, Kanban, Lean usw. umsetzen**
- **Eine Reihe von Best Practices wie zum Beispiel kurze Iterationen, Zusammenarbeit bei der Planung, funktionsübergreifende Teams, Einbeziehung der Kunden, regelmäßige Retrospektiven und kontinuierliche Wertbeiträge.**

– Quelle: Forbes, „**The Agile Mindset, Revolutionizing Software Development Team**“

Produktteams:

- Realisieren Innovationen via Software und Hardware
- Arbeiten in sicherheitskritischen und regulierten Branchen
- Arbeiten mit diversen Zulieferern und Kunden zusammen
- Arbeiten an ambitionierten, großen Projekten
- Nutzen eine Plattformstrategie, um Kosten zu senken
- Verwalten Produktvarianten und kundenspezifische Besonderheiten

Darüber hinaus haben Produktteams viele Herausforderungen gemeinsam, unter anderem die kontinuierliche Weiterentwicklung der Vorschriften, verteilte Teams, außerordentliche Komplexität und ständiger Druck, Erstanbietervorteile zu erzielen.

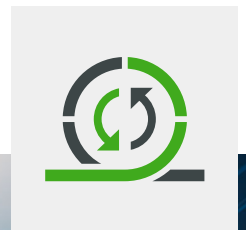
Wie können sich diese Teams die zahlreichen Vorteile agiler Entwicklung in einer stärker skalierbaren, disziplinerteren und besser überprüfbareren Art und Weise zunutze machen? Hier kommen hybrid-agile Frameworks ins Spiel – manchmal unter dem Begriff „planbasierte agile Prozesse“ bekannt.

Hybrid-agile Prozesse verbinden agile Entwicklungstechniken mit zusätzlichen Elementen, um diese für die Produktentwicklung in Medizin, Automobilbau, Luft- und Raumfahrt sowie weiteren Industriebranchen praktischer und skalierbarer zu gestalten.

PTC hatte das Privileg, mit zahlreichen weltweit führenden Fertigungsunternehmen bei ihrer Transition oder der Automatisierung ihrer skalierten, agilen Entwicklungsprozesse zusammenzuarbeiten. Viele Kunden implementieren unsere Software, um ihre Produktentwicklungsprozesse zu optimieren oder zu automatisieren. Andere nutzen sie, um eine umfassende Transition zu einem neuen und agileren Prozess zu unterstützen.

Wie jede Neuerung kann auch die Prozesstransformation zur Herausforderung werden. Unterwegs tappen die Teams mitunter in Fallen und wären im Nachhinein froh gewesen, wenn jemand sie rechtzeitig gewarnt hätte. Dieses E-Book ist das Ergebnis von Lektionen, die wir dank unserer engen Zusammenarbeit mit Hunderten von Produktteams gelernt haben, die den Übergang zu einem skalierten, agilen Entwicklungsprozess erfolgreich bewältigt haben.

Wir hoffen, dass unsere Erkenntnisse Ihnen bei Ihrer eigenen Transformation von Nutzen sein werden!



Die neun größten Fehler, die es zu vermeiden gilt

1

Konzeption
eines eigenen
Frameworks von
Grund auf

2

Widerstand auf
individueller Ebene

3

Widerstand auf
der Ebene der
Organisation

4

Fehlende ALM-
Funktionalitäten

5

Monolithische
Architektur

6

Fehlende
übergeordnete
Planung

7

Kein
allgemeingültiger
Prozess

8

Kein aussagefähiges
Kunden-Feedback

9

Kontinuierlicher
schleichender
Funktionszuwachs

1/

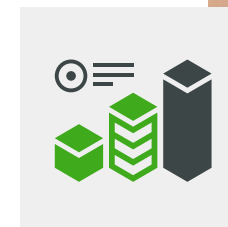
Konzeption eines eigenen Frameworks von Grund auf

Jedes Unternehmen ist einzigartig – mit eigenen, individuellen Anforderungen, Kunden und Marktbedingungen.

Angesichts dessen ist es für manche Organisationen womöglich verlockend, auf dem Weg zu einem skalierten, agilen Entwicklungsprozess ein eigenes agiles Framework zu erarbeiten. Besonders groß ist die Versuchung für die Zuständigen innerhalb der Organisation, einfach ein paar agile Elemente an bestehende Wasserfallprozesse anzuhängen und die Aufgabe als erledigt zu betrachten.

Diese DIY-Methode ist zwar möglich, aber aus unserer Sicht nicht empfehlenswert. Stattdessen empfehlen wir, mit einem etablierten hybrid-agilen Framework zu beginnen und dieses an die Anforderungen der eigenen Organisation anzupassen. Mit einer hochwertigen agilen Vorlage zu beginnen, hat viele Vorteile

- Schnellere Amortisierung
- Nutzung bewährter Frameworks, die den Test der Zeit bereits bestanden haben
- Verfügbarkeit von Schulungs-, Dokumentations-, Community- und Beratungsressourcen
- Problemlose Automatisierung, da Frameworks in Toolsets wie PTC Codebeamer instantiiert werden können



Wo sollte man anfangen? Forscher an der Universität von Cambridge haben **25 hybride Strategien** herausgearbeitet, die die Stärken agiler und plangesteuerter Entwicklung miteinander verbinden. Es würde die Grenzen dieses E-Books sprengen, alle diese Strategien zu beschreiben. Daher wollen wir im Folgenden nur drei skalierbare, agile Modelle darstellen, die wir am häufigsten sehen:

Scaled Agile Framework (SAFe)

wurde von Dean Leffingwell entwickelt, einer führenden Autorität zum Thema agiler Entwicklung. Bei diesem skalierten, agilen Framework handelt es sich um einen hochgradig präskriptiven Prozess, der den Wertbeitrag auf drei Ebenen optimiert: Portfolio, Programm und Team. Dieses Framework wurde 2011 vorgestellt, seither kontinuierlich aktualisiert und steht jetzt unter www.scaledagileframework.com öffentlich zur Verfügung.

Large Scale Scrum (LeSS)

wurde 2005 von Craig Larman und Bas Vodde im Rahmen ihrer Arbeit bei Nokia Siemens Networks entwickelt. LeSS definiert zwei Frameworks: ein kleines Modell, skalierbar für bis zu 64 Personen, und LeSS Huge, skalierbar für Tausende von Personen. Es ist weniger präskriptiv als SAFe und stellt einen eleganten Weg dar, den Projektaufwand zu minimieren.

Disciplined Agile Delivery (DAD)

wurde 2012 von Scott Ambler und Mark Lines entwickelt und 2019 vom Project Management Institute, einem gemeinnützigen Verband, erworben. DAD ist nicht präskriptiv und bietet sechs verschiedene Versionen des Bereitstellungszyklus, wobei es sich auf agile, Lean, Kanban, Scrum und andere etablierte Agilitätstechniken stützt.

Im Lauf der Zeit werden Sie Ihr Modell höchstwahrscheinlich an die individuellen Anforderungen Ihrer Organisation anpassen. Dazu ein Wort der Warnung: Widerstehen Sie der Versuchung, Verfahren hinzuzufügen, „weil wir das schon immer so gemacht haben“. Dies ist für Ihre Organisation die Chance auf Veränderung. Wenn Sie sich an die agilen Prinzipien kontinuierlicher Iterationen und Verbesserungen halten, können Sie sich darauf verlassen, dass letztendlich ein Prozess entsteht, der auf Ihre Organisation passt – ganz gleich, was Ihr Ausgangspunkt war.

2/

Widerstand auf individueller Ebene

Veränderungen bedeuten immer ein Risiko. Bei Einzelpersonen und Unternehmen, die auf Risikoaversion hin geprägt wurden, kann der Wechsel zu einem neuen Produktentwicklungsprozess – ganz gleich, wie vielversprechend – Ängste, Ressentiments und aktiven Widerstand hervorrufen.

Widerstand gegen Veränderungen manifestiert sich bei Einzelpersonen beispielsweise durch Beschwerden, Blockaden und offene Freude über kleine Schwierigkeiten, wie sie mit jeder Veränderung einhergehen.

Wie lässt sich Widerstand auf individueller Ebene überwinden?

Ein Grundgesetz der menschlichen Natur lautet: Je stärker die Kraft, die man aktivem Widerstand entgegensetzt, desto heftiger der Gegendruck. Zwingt man Menschen zu Veränderungen, so beharren sie oftmals umso stärker auf ihrer Position. Wir empfehlen Ihnen, die Aufmerksamkeit stattdessen einfach auf andere Dinge zu richten.

Den für die Transformation zuständigen Führungskräften empfehlen wir Folgendes:

1. Erläutern Sie das Warum. Stellen Sie die Vorteile der Agilität und die Gründe, warum Ihre Organisation eine Transformation in Angriff nimmt, im Detail dar.
2. Versuchen Sie, Menschen in Rollen einzusetzen, die sie mögen und in denen sie Erfolg haben werden.
3. Finden Sie heraus, wer Ihre internen Unterstützer sind – also Teammitglieder, die offen für Veränderungen sind (häufig „Early Adopter“ genannt) und die sich von den Vorteilen neuer Arbeitsverfahren motivieren lassen – und setzen Sie diese Unterstützer bei agilen Projekten in Rollen ein, die für andere deutlich wahrnehmbar sind.
4. Feiern Sie die Erfolge agiler Teams.

Wenn die anderen Teammitglieder sehen, dass neue Prozesse funktionieren und zum Erfolg führen, lässt der Widerstand im Lauf der Zeit normalerweise nach.

3/ Widerstand auf der Ebene der Organisation

Laut der unabhängigen, gemeinnützigen Scrum Alliance erleben über **70 % derjenigen, die Agilität umzusetzen versuchen**, Spannungen zwischen ihren Teams und der übrigen Organisation.

Organisationen, die sich gegen Veränderungen sträuben, erkennt man beispielsweise an unrealistischen Erwartungen, viel zu engen Zeitplänen oder schwankendem Engagement für neue Prozesse.

In einer idealen Welt schafft die Unternehmensleitung ein Umfeld, in dem Veränderungen und Risikofreude akzeptiert und unterstützt werden und das Lernprozesse begünstigt. Mit folgenden Tipps lässt sich Widerstand vonseiten der Organisation abbauen:

1. **Beginnen Sie mit dem richtigen Projekt.** Wählen Sie ein Pilotprojekt, das für Ihr Unternehmen sinnvoll, aber im Fall eines Fehlschlags nicht existenzbedrohend ist. Das verschafft dem Pilotteam den nötigen Freiraum, um mit neuen Arbeitsverfahren zu experimentieren.
2. **Definieren Sie realistische Erwartungen.** Kurze agile Sprints bieten die Chance, den Verlauf der Umsetzung zu überprüfen sowie Korrekturen und Anpassungen vorzunehmen. Das liegt in der Natur der Sache und ist der Schlüssel zur agilen Herangehensweise. Auf Stakeholder aber, die damit nicht vertraut sind, machen die ersten Sprints im Vergleich zur Prüfung fertiger Produkte, wie sie sie aus der Vergangenheit kennen, oftmals wenig Eindruck. Bereiten Sie wichtige Stakeholder darauf vor, im Idealfall durch Schulungen und Weiterbildungsmaßnahmen, damit sie wissen, was sie erwartet, und nicht überrascht sind.
3. **Feiern Sie Erfolge.** Erfolge gibt es bei agilen Projekten in vielen Varianten – vom ersten Kickoff eines Pilotprojekts bis zur ersten freigegebenen Version. Feiern Sie eine Party, hängen Sie ein Banner auf, rufen Sie es von den Dächern!



4. **Definieren Sie KPIs und verfolgen Sie diese nach.** Denn wenn man nicht weiß, wohin es gehen soll, dann landet man meistens woanders, wie es so schön heißt. Es empfiehlt sich, KPIs (Leistungskennzahlen) im Rahmen einer agilen Transformation gleich von Anfang an nachzuverfolgen. Auf diese Weise kann die Organisation die Gründe, weswegen sie sich überhaupt auf eine Transition eingelassen hat, deutlicher machen, und die Fortschritte auf dem Weg zu ihren Zielen klarer darstellen. Überlegen Sie sich für nicht direkt messbare Attribute wie Kundenzufriedenheit KPIs, die mit diesen Attributen korreliert sind und gemessen werden können. Wir empfehlen, sich auf wenige KPIs zu konzentrieren. Im Folgenden finden Sie eine Liste möglicher KPIs.

| Attribute | Mögliche KPIs |
|--------------------------------------|--|
| Kundenzufriedenheit | Umfrageergebnisse, Produktverkäufe, Produktbewertungen |
| Produktqualität | Anzahl der Fehler/Anrufe beim Support/Rücksendungen, CAPAs nach der Freigabe |
| Entwicklungseffizienz | Anzahl der pro Sprint oder Epic erkannten Fehler (früher ist besser) |
| Time-to-Market | Dauer des Produktzyklus von der Konzeption bis zur Freigabe |
| Wiederverwendung | Anzahl der wiederverwendeten Module/Code-Zeilen |
| Projektspezifische Kennzahlen | Aussagefähige Kennzahlen mit Bezug zum Kundennutzen |

4/ Fehlende ALM-Funktionalitäten

Agile-Methoden werden häufig mit ALM (Anwendungslebenszyklus-Verwaltung) verwechselt. Es handelt sich jedoch um verschiedene, wenn auch miteinander verwandte Disziplinen.

Als agil bezeichnet man eine Methode in der Softwareentwicklung, die das Wer, Was, Wo und Wann des Softwareentwicklungsprozesses beschreibt.

ALM dagegen ist der strategische Prozess der Verwaltung eines Software- oder Produktlebenszyklus von der ersten Idee über Konzeption, Konstruktion, Entwicklung, Test und Bereitstellung bis hin zum Ende der Nutzungsdauer. ALM unterstützt alle Softwareentwicklungsprozesse einschließlich der agilen.

ALM ermöglicht Software-Engineering-Teams die effiziente Zusammenarbeit an Projekten, gestützt auf vertrauenswürdige, aktuelle Informationen. ALM-Funktionalitäten sind besonders wichtig für sicherheitskritische Entwicklungen, für deren Erfolg es vor allem auf Compliance ankommt. ALM ist eine grundlegende Disziplin für erfolgreiche Produkte, Teams und Unternehmen.

Fehlen klare ALM-Funktionalitäten, erleben die Teams häufig Herausforderungen wie die folgenden:

- Änderungen, die aufgrund mangelhafter Check-in-Hygiene verloren gehen
- Qualitätsprobleme aufgrund mangelhafter Versionskontrolle und inkonsistenter Testabdeckung
- Geringe Produktivität, da die Teammitglieder häufig nach „den richtigen“ Benutzerangaben, Codes und Testergebnissen suchen müssen
- Hohe Abhängigkeit von wichtigen Beschäftigten, die „wissen, wo alles ist“
- Niedrige Wiederverwendungsraten



Der Übergang zu einem agilen oder hybrid-agilen Prozess bietet die wunderbare Chance, eine Bestandsaufnahme der ALM-Funktionalitäten in der Organisation durchzuführen und bei Bedarf Änderungen vorzunehmen. Wie gut unterstützt Ihr derzeitiges ALM-Toolset Ihre Organisation?

Im Folgenden finden Sie eine Checkliste der ALM-Funktionalitäten mit direkten Auswirkungen auf den Erfolg:

| | | | |
|---|---|---|---|
| ✓ | Benutzerfreundlichkeit | ✓ | Alle Artefakte umfassende End-to-End-Nachverfolgbarkeit |
| ✓ | Kanban-Ansichten | ✓ | Mehrere Ansichten des gleichen filterbaren Rückstands |
| ✓ | Zusammenarbeit und absolute Transparenz jederzeit und überall | ✓ | Modernste Berichts-, KPI- und Dashboard-Funktionalitäten |
| ✓ | Lückenlose, nicht editierbare Prüftrails für Compliance-Zwecke | ✓ | Modernste Sicherheits- und Berechtigungssteuerung |
| ✓ | Hosting in der Cloud oder hybrid | ✓ | Integrationen mit Drittanbietersystemen und REST-APIs |
| ✓ | Funktionalitäten für die Wiederverwendung, u. a. Zusammenführung und Verzweigung | ✓ | Funktionalitäten für die Produktlinienentwicklung |
| ✓ | Integrationen mit PLM-, Systemmodellierungs- und Versionskontrollsystemen | ✓ | Unterstützung für OSLC (Open Services for Lifecycle Collaboration) |

5/ Monolithische Architektur

Die besten agilen Teams sind klein. Studien zeigen, dass Gruppen mit 3-8 Mitgliedern signifikant produktiver sind als Gruppen mit 9 und mehr Mitgliedern.

Wenn Teams gemeinsam an komplexen Produkten arbeiten, wird die Arbeit in Teilbereiche zerlegt. So können Teams das richtige Abstraktionsniveau für eine optimale Teameffizienz erreichen.

Für große Projekte sollten Organisationen eine locker verknüpfte Architektur mit klar definierten Schnittstellen anstreben. So können Module unabhängig entwickelt und getestet werden, ohne dass technische „Schulden“ in Form langwieriger Regressionstests in der Endphase des Projekts anfallen.

Mithilfe folgender Tipps können sich Teams mit ihren diversen Arbeitsgebieten aufeinander abstimmen. Die Tipps sind besonders wichtig für Teams, die mit einer monolithischen Architektur zurechtkommen müssen, zum Beispiel bei der Arbeit an Upgrades älterer Produkte, die aus Kostengründen nicht neu konzipiert werden können.

- Sorgen Sie für universellen, transparenten Zugriff auf die neueste Version der laufenden Arbeiten. Dies erfordert ein diszipliniertes Änderungsmanagement, bei dem ausschließlich autorisierte Benutzer Aktualisierungen festschreiben dürfen. Durch Zugriff auf den neuesten Quellcode können Teams „Informationssilos“ überwinden, die andernfalls schwierige, langwierige Integrationen nach sich ziehen würden.
- Sorgen Sie für eine strenge Governance in Bezug auf die Schnittstellen. So lässt sich gewährleisten, dass der Hack, der dem einen Team drei Monate Arbeit spart, nicht zum Alles-auf-Anfang-Moment für ein anderes Team wird.
- Testen Sie Integrationen frühzeitig und häufig – im Idealfall bei jedem Sprint (bei kleineren Projekten) bzw. jedem Epic (bei größeren Projekten).



6/ Fehlende übergeordnete Planung

Skalierte Agile-Methoden ermöglichen es Teams, mithilfe übergeordneter Planung umfangreiche, ambitionierte Projekte abzuwickeln.

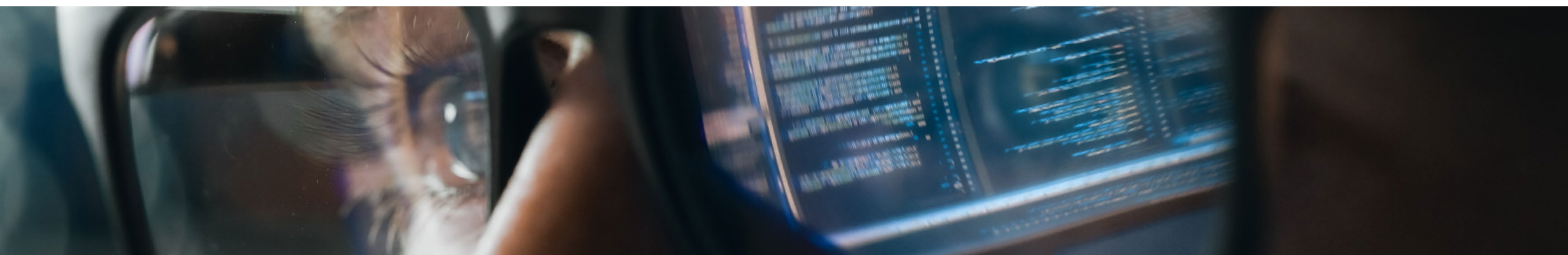
In der Tat gilt: Je größer das Projekt, desto mehr Planung ist notwendig, um sicherzustellen, dass die Teams auf die gleiche Produktvision hinarbeiten, wobei die Schnittstellen zwischen den Teams klar definiert sein müssen. Einzelne Scrum-Teams können durchaus das „Wie“ bestimmen, nicht jedoch das angestrebte „Was“.

Symptome für eine unzureichende übergeordnete Planung sind beispielsweise:

- Mehrere Sets widersprüchlicher Anforderungen oder Benutzerangaben
- Langwierige und schwierige Integrationen
- Sehr viel Ausschuss und Nacharbeit
- Abgeschlossene Projekte, die die übergeordneten Unternehmensziele nicht erfüllen

Es gibt zahlreiche Strategien für eine übergeordnete Planung. Bei manchen Frameworks wie zum Beispiel SAFe werden bestimmte Zuständigkeiten auf Portfolio-, Programm- und Teamebene zugewiesen, um die Projektplanung zu koordinieren. Andere Frameworks geben flexiblere Leitlinien vor und überlassen die Einzelheiten der Organisation.

Wenn Sie vermuten, dass es in Ihrer Organisation an übergeordneter Planung fehlt, ist es unter Umständen hilfreich, Ideen der stärker präskriptiven Leitlinien zum Thema aus dem SAFe-Framework zu übernehmen. Ein strengerer Planungsprozess kann sehr wohl zu erfolgreicheren Produktversionen beitragen.



7/ Kein allgemeingültiger Prozess

Selbststeuerung ist ein Schlüsselprinzip agiler Teams. Agile Teams verwalten ihr eigenes Arbeitspensum, weisen Aufgaben je nach Bedarf und Fähigkeiten neu zu und sind aktiv in wichtige Entscheidungen eingebunden.

Im Bestreben nach kontinuierlichen Verbesserungen sind manche Teams versucht, ihre eigenen Prozessvarianten zu kreieren. Dies mag für das Team gut funktionieren, ist aber auf Organisationsebene der Weg ins Desaster, denn mehrere Prozessvarianten:

- Schaffen Verwirrung im Hinblick auf Rollen und Zuständigkeiten
- Behindern den Austausch von Projektdaten und die Wiederverwendung
- Machen eine funktionierende Governance praktisch unmöglich, da die Projektkennzahlen für unterschiedliche Teams Unterschiedliches bedeuten

Heterogene Organisationen, in denen einige Teams nach agilen und andere nach Wasserfallprinzipien arbeiten, haben natürlich mindestens zwei Prozessvarianten. Unzulässige Varianten für bestimmte Teams sind ebenfalls möglich. Aber auch ein solches Szenario ist meilenweit entfernt von einer Organisation, in der „tausend verschiedene Blumen blühen“, also jedes Team nach seiner eigenen Prozessvariante arbeitet.

Selbstverständlich darf ein Prozess nicht statisch sein. Vielmehr sollte er sich mit wachsender Erfahrung in der Organisation und neuen Verbesserungsmöglichkeiten weiterentwickeln. Der entscheidende Punkt besteht darin, dass Organisationen diese Veränderungen proaktiv verwalten, kommunizieren und einführen.



8/

Kein aussagefähiges Kunden-Feedback

Der Projekterfolg hängt immer ganz entscheidend von aussagefähigem Kunden-Feedback ab.

Bei Wasserfallprojekten wird das Feedback in erster Linie am Anfang des Projekts bei der Erfassung der Anforderungen berücksichtigt. Da es während des Projektzyklus keine Feedback-Schleife gibt, kommt dieses in der Regel zu spät, um das Projektergebnis noch zu beeinflussen.

Bei agilen Projekten trifft Feedback während der gesamten Projektdauer kontinuierlich ein, da die Produktverantwortlichen kontinuierlich mit den Kunden kommunizieren, um geschäftliche Anforderungen abzuklären, den Rückstand zu bearbeiten und den Fortschritt am Ende jedes Sprints zu überprüfen. Diese kontinuierliche Feedback-Schleife ist einer der entscheidenden Unterschiede – und einer der wichtigsten Vorteile – der agilen Herangehensweise.

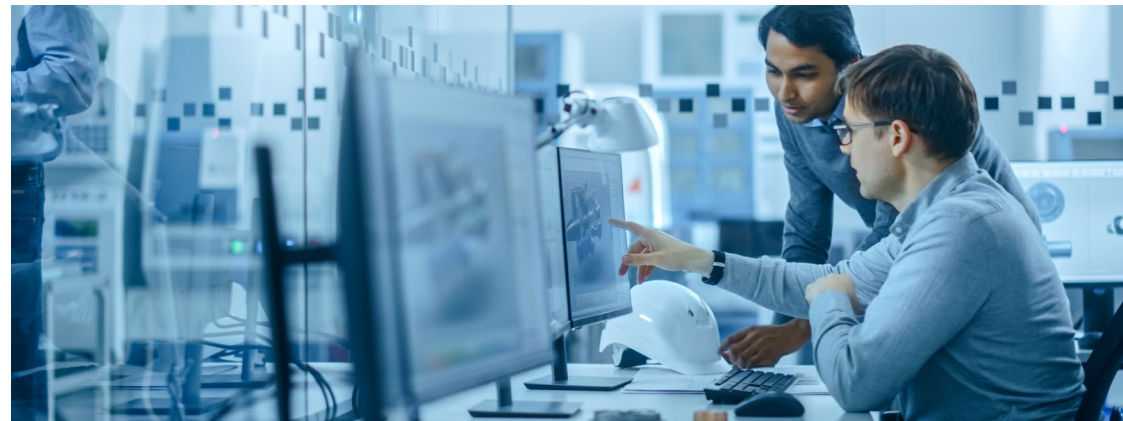
Wie die Softwareentwicklung selbst ist auch eine exzellente Kundenkommunikation Kunst und Wissenschaft zugleich. Um aussagefähiges Feedback einholen zu können, muss der Produktverantwortliche:

- Mit dem Geschäft und seinen Anforderungen vertraut sein
- Klug genug sein, um bei Bedarf Hilfe zu holen
- Verfügbar sein, um das Team in seiner Tätigkeit zu unterstützen

Symptome für unzureichendes Kunden-Feedback sind unter anderem:

- Projekte werden genehmigt und abgeschlossen, aber nie implementiert.
- Projekte werden in sehr späten Phasen noch einmal stark überarbeitet und verändert.
- Bei Projekten kommt es aufgrund von fehlendem Input ständig zu Verzögerungen oder sie werden unterbrochen.
- Projektteams können ihre Ressourcen aufgrund von fehlendem Input nur unzureichend nutzen.

Letztendlich ist es Aufgabe des Projektponsors, dafür zu sorgen, dass Kundenvertreter in den Teams tätig werden, die das Wissen, die Weisheit und die nötige Verfügbarkeit aufbringen, um das Projekt zum Erfolg zu führen. Wenn Sie Lücken bei diesen kritischen Zuständigkeiten vermuten, empfehlen wir Ihnen, sensibel und diplomatisch mit dem Produktverantwortlichen, dem Sponsor oder beiden zusammenzuarbeiten, um das Problem zu beheben.



9/

Kontinuierlicher schleichender Funktionszuwachs

Bei einem agilen Prozess können Teams Projekte in Angriff nehmen, ohne bereits alle Antworten zu kennen. Stattdessen verfeinern die Teams im Rahmen eines iterativen Prozesses im Lauf der Zeit die Antworten und nähern sich auf diese Weise dem erfolgreichen Projektabschluss in Form des fertigen Arbeitsergebnisses.

Häufig hat dieses Endergebnis kaum noch Ähnlichkeit mit dem anfänglichen Prototyp. Das ist alles andere als ein Problem. Vielmehr ist es ein Zeichen, dass der Prozess so funktioniert, wie er soll, indem das Kunden-Feedback nämlich zeitnah umgesetzt wird. Bei jeder Iteration einer funktionierenden Software ergeben sich neue Fragen und Verbesserungschancen.

Aber Vorsicht: Die „agile Flexibilität“ des einen Teams ist der „schleichende Funktionszuwachs“ des anderen. Erfolgreiche agile Teams durchlaufen vielleicht eine oder zwei experimentelle oder eigenwillige Iterationen. Im Lauf der Zeit sollte sich ein Projekt aber zu einem stetigen Marsch in Richtung einer Lösung entwickeln.

Das Funktionszuwachsproblem tritt auf, wenn Projektverantwortliche auch nach mehreren Iterationen noch kein klares Bild der Lösung vor Augen haben und die Kundenvertreter stattdessen einen Versuchsballon nach dem anderen steigen lassen, ohne dass die endgültige Lösung je näherrückt. In diesem Fall entsteht bei jeder Iteration aufs Neue der Eindruck, man stehe wieder ganz am Anfang.

In der Regel kommt es zu diesem Problem, wenn Teams an einer nicht eindeutig definierten Lösung arbeiten, die eine neue oder erst im Entstehen begriffene Technologie oder Marktnische betrifft, und die Anforderungen nur sehr grob umrissen sind. Statt das Problem gründlich zu untersuchen, wird eine Entscheidung getroffen, um möglichst schnell etwas Auslieferbares in der Hand zu haben.

Wenn es bei einem Projekt zu einem kontinuierlichen schleichenden Funktionszuwachs kommt, sollten Sie bei der technischen Entwicklung neue Prioritäten setzen und das Problem aus einer neuen Perspektive angehen. Iterationen, die sich gezielt auf Marktforschung, Fokusgruppen und Wettbewerbsanalysen beziehen, eignen sich als bewährte Methoden, um die Ausrichtung auf einen Lösungsvorschlag rasch voranzutreiben.

In manchen Organisationen liegen solche Untersuchungen in der Hand eines anderen Teams. Ausgestattet mit neuen Erkenntnissen können die Entwicklerteams dann Iterationen erarbeiten, die letztendlich die Grundlage einer Lösung bilden.

Über PTC Codebeamer

Codebeamer ist eine moderne, integrierte, vernetzte ALM-Lösung für die Produkt- und Softwareentwicklung.

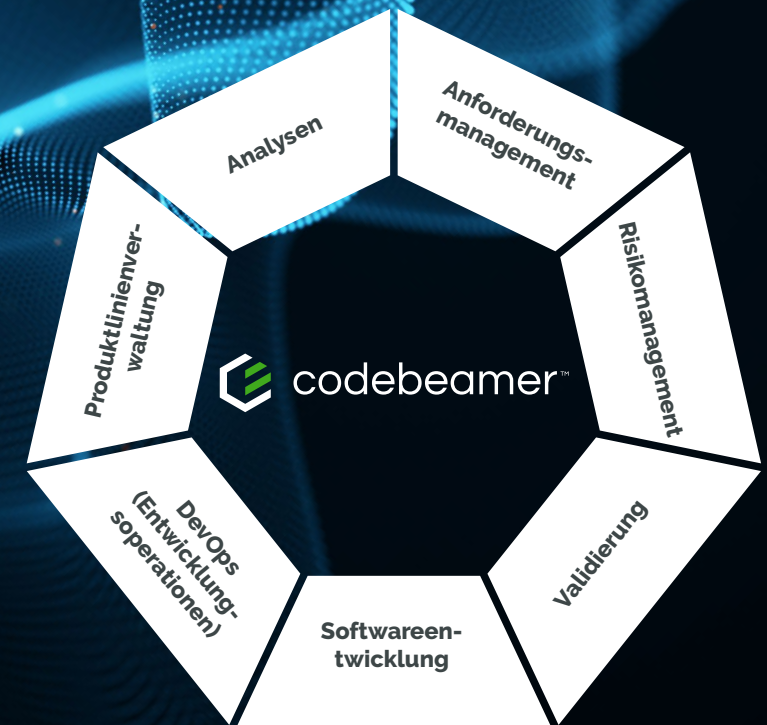
Sie erweitert ALM um Funktionen zur Konfiguration von Produktlinien und bietet eine klare Konfigurierbarkeit für komplexe Prozesse. Codebeamer ermöglicht durch digitale Workflows mehr Effizienz bei der Zusammenarbeit in der Entwicklung und unterstützt die Einhaltung der Vorschriften. Codebeamer ist mit anderen Produkten von PTC integrierbar, sodass die technische Entwicklung mit in den Digital Thread eingebunden werden kann.

Codebeamer erleichtert durch digitale Workflows die Zusammenarbeit in der Entwicklung sowie die Einhaltung der Vorschriften und ermöglicht eine effiziente Produktlinienentwicklung. Sämtliche Entwicklungs-Tools können miteinander verknüpft werden, um einen zentralen Entwicklungs-Hub für alle Teams zu schaffen. Die Lösung lässt sich problemlos an spezifische Anforderungen der technischen Entwicklung anpassen und ermöglicht eine automatisierte Prozesskontrolle zur Einhaltung regulatorischer Vorschriften.

Codebeamer schafft die Grundlage für Klarheit und effiziente Zusammenarbeit im Umgang mit agilen Iterationen und Produktversionen.

- Unterstützung für skalierte agile Prozesse (LeSS, DAD, SAFe®)
- Geeignet für Scrum, Kanban, XP und jede andere Lean-/Agile-Methode
- Visualisieren und Verwalten von Aufgaben mittels anpassbarer Kanban-Tafeln
- Analyse agiler Leistung mittels Burndown- und anderer KPI-Diagramme

Weitere Informationen finden Sie unter www.ptc.com/codebeamer.





DIGITAL TRANSFORMS PHYSICAL

121 Seaport Blvd, Boston, MA 02210, USA: ptc.com/de

© 2023, PTC Inc. Alle Rechte vorbehalten. Die Inhalte dieser Seiten werden ausschließlich zu Informationszwecken bereitgestellt und beinhalten keinerlei Gewährleistung, Verpflichtung, Bedingung oder Angebot seitens PTC. Änderungen der Informationen vorbehalten. PTC, das PTC Logo und alle anderen PTC Produktnamen und Logos sind Warenzeichen oder eingetragene Warenzeichen von PTC und/oder Tochterunternehmen in den USA und anderen Ländern. Alle anderen Produkt- oder Firmennamen sind Eigentum ihrer jeweiligen Besitzer.

100-2023-Skalieren-agiler-Entwicklung-neun-Fehler-die-es-zu-vermeiden-gilt-08-21-de