
Klassisches Änderungsmanagement für agile IT-Projekte?

Version v004 vom 29.07.2021 / final

Frank Sarre

Projective Expert Group

Christian Welkenbach

TCI Rechtsanwälte

Herbstakademie 2021

Agenda

1. Einleitung

2. Problemaufriss

- 2.1 Begriffe und Abgrenzungsfragen
- 2.2 Unvermeidbarkeit von Änderungen
- 2.3 Leistungsänderungen in klassischen IT-Projekten
- 2.4 Vertragliche Regelungen für Change Requests
- 2.5 Leistungsänderungen in agilen IT-Projekten
- 2.6 Probleme mit aktuellen agilen Vorgehensmodellen

3. Lösungsansatz und kritische Würdigung

- 3.1 Möglicher Lösungsansatz für agile IT-Projekte
- 3.2 Mögliche Kritikpunkte

4. Fazit

1. Einleitung (I)

- ▶ Weiterhin starker Trend, komplexe IT-Projekte mit **agilen Projektmethoden** durchzuführen
- ▶ **Wasserfall-basierte Projektmethoden** treten zunehmend in den Hintergrund
- ▶ Agile Methoden bieten **viele Vorteile** für Kunden und Dienstleister, u.a. wird mit folgenden Punkten geworben:
 - ❖ Effiziente Projektdurchführung
 - ❖ Klare Fokussierung auf das Projektziel
 - ❖ Die gesamte Fachlichkeit der Anwendung muss erst im Laufe des Projekts festgelegt werden
 - ❖ Geringere Aufwände insgesamt
 - ❖ Geringere Kosten
 - ❖ Kürzere Projektlaufzeit
 - ❖ Hohe Softwarequalität

1. Einleitung (II)

- ▶ In der Praxis sind Kunden jedoch **häufig unzufrieden** mit den Ergebnissen ihres agilen Projekts
- ▶ Manche Projekte werden sogar **als gescheitert** eingestuft
- ▶ Die **Gründe** für das Scheitern eines agilen Projekts sind **vielfältig**
- ▶ Häufig explodiert das **Budget**, ohne dass die zu entwickelnde Software am Ende des Projekts einen einsatzfähigen Stand erreicht hätte

2. Problemaufriss

2.1 Begriffe und Abgrenzungsfragen

- ❖ Änderungen: „**Changes**“
- ❖ Änderungsverlangen: „**Change Requests**“ (CRs)
- ❖ Was soll geändert werden können und **welche Auswirkungen** hat die jeweilige Änderung auf den **Projektplan**?
- ❖ Was ist seitens des Auftraggebers **zu vergüten** und was nicht?
→ Siehe nächste Folie

- ❖ **Problempunkt:**
In der Praxis gibt es bei fehlender Spezifikation **großes Streitpotential**

2. Problemaufriss (fortgesetzt)

Hinsichtlich CRs erscheint folgende Differenzierung hilfreich:

Nr.	Art der Anforderung bzw. Leistung	Zusätzliche Vergütung	Anmerkung
1	„Echt“ neu	ja	(„Echter“) CR (Zusatzfunktionalität)
2	Leistungskonkretisierung	nein	Kein CR, nur Präzisierung von bereits definierten Leistungen
3	Leistungsänderung	ja (ggf. auch Reduzierung)	(„Echter“) CR, Austausch bestehender Leistungsteile
4	Mängelbeseitigung	nein	Kein CR, Mängel müssen im Rahmen der Sach- und Rechtsmängelhaftung kostenfrei beseitigt werden
5	Service Request und ähnliche Leistungen	ja	Kein CR, im Regelfall eine Dienstleistung

2. Problemaufriss (fortgesetzt)

2.2 Unvermeidbarkeit von Änderungen

- ❖ Auftraggeber / Kunden sind nur selten in der Lage, ohne Kenntnis der technischen Möglichkeiten das gewünschte Gesamtsystem zu einem frühen Zeitpunkt im Projekt vollständig zu spezifizieren
 - In der Praxis sind Änderungen kaum zu vermeiden

2.3 Leistungsänderungen in klassischen IT-Projekten

- ❖ Wasserfallmodell-basierte Projekte haben den Vorteil einer **dedizierten Spezifikationsphase**
 - Abgestimmtes **Pflichtenheft** → **transparente Leistungen**
 - **Ständige Fortschreibung** bei Änderungen, wenn CRs entsprechend gehandhabt werden
 - **Geschuldete Leistung** ist zu jedem Zeitpunkt durch die ursprüngliche Spezifikation und die nachträglichen CRs **definiert**

2. Problemaufriss (fortgesetzt)

2.4 Vertragliche Regelungen für Change Requests

- ❖ (In IT-Projekten aller Art) herrscht ein hohes Interesse der Vertragspartner an **belastbaren CR-Regelungen**
- ❖ Üblicherweise wird ein CR-Verfahren **mit folgenden Inhalten** vereinbart:
 - Wer darf Änderungen verlangen?
 - Ablauf / formalisiertes Procedere
 - Prüfungsaufwand und Vergütung
 - Folgen (Spezifikation, Tests, Vergütung, Projektplan, ...)
 - Mitwirkungsleistungen (des Auftraggebers)
 - Fristen für Verfahrensschritte
 - Eskalation & Konfliktlösung

2. Problemaufriss (fortgesetzt)

2.5 Leistungsänderungen in agilen IT-Projekten

- ❖ **Keine initiale (Gesamt-) Spezifikationsphase** – das Projektziel wird anfangs nur grob definiert und dann in Iterationen immer weiter konkretisiert bzw. erweitert
- ❖ Jede Anforderung ist entweder **neu** oder eine **Änderung** der bestehenden Anforderungen
- ❖ Pro Iteration / Sprint wird eine festgelegte Menge **priorisierter Anforderungen** umgesetzt
- ❖ **Änderungen / Erweiterungen des Leistungsumfangs** sind in agilen IT-Projekten ausdrücklich erwünscht bzw. „systemimmanent“
- ❖ **Ohne permanente Fortschreibung der Spezifikation und der Abnahmekriterien** kann die Kontrolle über das Projekt abhanden kommen
 - Möglicherweise grobe Meinungsunterschiede über den Leistungsumfang
 - Heftige Budgetüberschreitung nicht unwahrscheinlich
 - Software im Ergebnis wahrscheinlich nicht einsatzfähig
 - Mängelrügen fast unmöglich

2. Problemaufriss (fortgesetzt)

2.6 Probleme mit aktuellen agilen Vorgehensmodellen

- ❖ Der Auftraggeber erhält zu jeder einzelnen fachlichen Anforderung **zu spät** eine Rückmeldung über:
 - finanzielle,
 - personelle,
 - fachliche,
 - technische und
 - organisatorische

Auswirkungen der jeweiligen Anforderung.

→ Genau dies wäre aber **für die Freigabe** jeder einzelnen Anforderung erforderlich.
- ❖ Alle möglichen Arten von Leistungsänderungen werden in agilen Projekten üblicherweise **vermischt** und pauschal nur als **vergütungspflichtige** Anforderungen / Änderungen / Change Requests deklariert
 - **Schwierige Budgetkontrolle**

3. Lösungsansatz und kritische Würdigung

3.1 Möglicher Lösungsansatz für agile IT-Projekte

Für jede einzelne Anforderung / Änderung ist zu bewerten,

- was die Anforderung / Änderung **kosten** wird,
- welche **organisatorischen Auswirkungen** sich auf der Seite des Auftraggebers ergeben,
- welche **Fertigstellungstermine** zu verschieben sind,
- wie hoch die **Aufwände** hinsichtlich der relevanten **Dokumentationen** ausfallen werden,
- ob **andere Anforderungen** oder deren Implementierungen fachlich oder technisch beeinflusst werden,
- welche Anpassungsmaßnahmen hinsichtlich **Applikationslandschaft** und **Infrastruktur** des Auftraggebers vorzunehmen sind.

3. Lösungsansatz und kritische Würdigung

3.1 Möglicher Lösungsansatz für agile IT-Projekte (fortgesetzt)

Schlussfolgerungen:

- ❖ Die einzusetzende agile Projektmethode muss **möglichst exakt definiert** werden (ähnlich wie bei einem klassischen CR-Verfahren!)
 - ❖ Im **Vertrag** sind entsprechende Vereinbarungen aufzunehmen
 - ❖ Die Umsetzung muss mit entsprechenden **Projektmanagementmaßnahmen** erfolgen
- Der Auftraggeber erhält die Möglichkeit, für jede einzelne Anforderung eine **fundierte Entscheidung** zu treffen (hinsichtlich Kosten und sonstigen Auswirkungen)
- Auf diese Art und Weise werden **Unklarheiten vermieden** und das **Streitpotential wird minimiert**

3. Lösungsansatz und kritische Würdigung

3.1 Möglicher Lösungsansatz für agile IT-Projekte (fortgesetzt)

Was für jede einzelne Anforderung zu tun ist:

- ❖ Ermittlung **aller fachlichen Details**, die für die vollständige fachliche Spezifikation der neuen Anforderung erforderlich sind
- ❖ Klärung im Hinblick auf die Vergütungsfrage, ob die neue Anforderung
 - **tatsächlich neu** ist oder ob eine oder mehrere bestehende **Leistungen geändert** werden sollen,
 - einer (bloßen) **Leistungskonkretisierung** entspricht,
 - eher eine **Mängelbeseitigung** darstellt oder ob
 - einem **Service Request** entspricht (oder ob eine sonstige Dienstleistungsanforderung vorliegt)

3. Lösungsansatz und kritische Würdigung

3.1 Möglicher Lösungsansatz für agile IT-Projekte (fortgesetzt)

❖ Des Weiteren ist zu tun:

- Definition von **Testfällen, Testdaten und Abnahmekriterien**
- Ausarbeitung des **technischen Designs** auf Basis der fachlichen Spezifikation der neuen Anforderung
- Ggf. Erweiterung des **Architekturkonzepts**
- Bestimmung aller **Aufwände**, die für die Umsetzung der Anforderung erforderlich sind
- Bestimmung der **organisatorischen Auswirkungen** für den Auftraggeber
- Analyse, ob **Meilensteine** des Projekts verschoben werden müssen und wenn ja, um welche Zeiträume
- Klärung der **Auswirkungen auf andere Arbeitspakete** (Anpassungen, Entfall von Arbeitspaketen oder vorläufiger Stopp gewisser Arbeiten an bestimmten Arbeitspaketen)
- Klärung der **Mitwirkungsleistungen des Auftraggebers**

3. Lösungsansatz und kritische Würdigung

3.1 Möglicher Lösungsansatz für agile IT-Projekte (fortgesetzt)

❖ Vorteile für den Auftragnehmer:

- Fachlich und technisch sind alle **Voraussetzungen für die erforderlichen Implementierungsarbeiten** geklärt
- Für jede Anforderung bzw. Änderung besteht eine hohe Chance, Realisierungsaufwände und Kosten verbindlich festzulegen.
- Der Auftragnehmer gerät durch die Bearbeitung der neuen Anforderung nicht weiter unter Druck, da **Auswirkungen auf andere Meilensteine** ermittelt und gegebenenfalls **neue Termine** vereinbart werden können.
- Die **Projektplanung** kann sachgerecht angepasst werden
- Im Zuge der Prüfung jeder Anforderung bzw. CRs sind bereits alle **Testerfordernisse** und **Abnahmekriterien** geklärt worden.
- Aufgrund der zu jedem Zeitpunkt bereits fixierten fachlichen und technischen Spezifikation können später auftretende **Fehler, Störfälle oder sonstigen Defizite** geklärt werden.
- **Andere Anforderungen** lassen sich aufgrund der aktuellen fachlichen Spezifikation klar **abgrenzen**.

3. Lösungsansatz und kritische Würdigung

3.2 Mögliche Kritikpunkte

Verfechter agiler Projektmethoden können auf folgende Punkte verweisen:

- ❖ **Weitere Verfahrensschritte** entsprechen nicht mehr dem ursprünglichen Gedanken eines „leichtgewichtigen“ Vorgehens
- ❖ Das **Zeitfenster für die Planung** einer Iteration / eines Sprints (in SCRUM) ist in aller Regel knapp bemessen
- ❖ Der hier vorgeschlagenen Lösungsansatz entspricht nicht dem **Prinzip einer „schlanken Vertragsverhandlung“** („customer collaboration over contract negotiation“)

ABER: Justiziabilität, Brauchbarkeit der Arbeitsergebnisse und der Dokumentation sowie die Planung des Projekts können bei Standardansätzen agiler Projektmethoden leiden

4. Fazit

- ▶ **Änderungen** sind in laufenden Projekten praktisch **unvermeidbar**
- ▶ **CR-Verfahren** liefern dem Auftraggeber in einem klassischen Projekt eine gute Grundlage, um über die jeweilige Änderung zu entscheiden
- ▶ **Agile Projektmethoden** bieten dem Auftraggeber in aller Regel **keine brauchbare Entscheidungsgrundlage**, um über die einzelnen Anforderungen zu entscheiden, da die Konsequenzen einer Anforderung bzw. Änderung kaum unübersehbar sind
- ▶ **Ein klassisches CR-Verfahren lässt sich vorteilhaft in eine agile Projektmethode integrieren**, um die angesprochenen Defizite zu eliminieren
→ Insgesamt mehr **Budgetkontrolle** und **höhere Chance auf Erfolg**
- ▶ **Die Vorteile des Lösungsansatzes dürften die Kritikpunkte im Ergebnis überwiegen**