



Annex zu den AEB für Erstellung von Software nach agilen Methoden („Agile Softwareentwicklung“)

Neben AEB gelten für die Erstellung von Software in agilen Projekten zusätzlich folgende Bestimmungen. Sofern nicht anders angegeben, haben die hier definierten Begriffe die gleiche Bedeutung wie in den AEB.

1. Definitionen

Agile Softwareentwicklung ist ein Rahmenwerk, innerhalb dessen Menschen komplexe adaptive Aufgabenstellungen angehen können. Es besteht aus agilen Teams und den zu ihnen gehörenden Rollen (Product Owner, Agile Master, Entwicklungsteam,...), Ereignissen (Sprint, Sprint Planning, Daily Agile Meeting, Sprint Review, Sprint Retrospective,...), Artefakten (Product Backlog, Sprint Backlog, Increment,...) und Regeln.

Ein **Agile Team** besteht aus dem Product Owner, dem Entwicklungsteam sowie dem Agile Master. Es ist selbstorganisierend und entscheidet selbst, wie es seine Arbeit am besten erledigt.

Der **Product Owner** ist dafür verantwortlich, den Wert des Produktes zu maximieren, und insofern für die Steuerung der Entwicklungsleistungen, die schrittweise Formulierung und Konkretisierung der geschäftlichen und funktionalen Erwartungen an ein Produkt gegenüber dem Entwicklungsteam. Der Product Owner ist die einzige Person, die für das Management, insbesondere die verbindliche Erstellung und Aktualisierung, des Product Backlogs verantwortlich ist. Er kann sich beim Management des Product Backlogs unterstützen lassen, bleibt jedoch immer rechenschaftspflichtig.

Der **Agile Master** ist dafür verantwortlich, agile Softwareentwicklung zu fördern und zu unterstützen. Agile Master tun dies, indem sie allen Beteiligten helfen, die Theorie, Praktiken, Regeln und Werte des Agilen Rahmenwerks richtig zu verstehen und zu nutzen, also vor allem auch die Sprints und deren Ereignisse zu organisieren. Der Agile Master ist dafür verantwortlich das Agile

Team bei der Erreichung der Ziele zu unterstützen und *Impediments* zu beseitigen. Hierzu zählt auch, die Einwirkungen des Umfelds auf das Agile Team zu steuern und Abstimmungen mit dem Umfeld zu koordinieren.

Das **Agile Entwicklungsteam** ist für die Verwaltung, Organisation und Durchführung aller Entwicklungsarbeiten verantwortlich, die erforderlich sind, um ein Inkrement zu erstellen. Es ist interdisziplinär und verfügt über alle Kompetenzen, die erforderlich sind, um die Arbeit zu erledigen.

Entwicklungsteams sind von der Organisation so strukturiert und befähigt, dass sie ihre eigene Arbeit selbst organisieren und managen.

Der **Sprint** ist ein feststehender Zeitraum [Time Box] von in der Regel maximal einem Monat Länge, innerhalb dessen ein fertiges ["Done"], nutzbares und potenziell auslieferbares Inkrement hergestellt wird. Alle Sprints innerhalb eines Entwicklungsvorhabens sollten die gleiche Dauer haben. Der neue Sprint startet sofort nach Abschluss des vorherigen Sprints. Ein Sprint beinhaltet und umfasst alle agilen Tätigkeiten und Ereignisse, insbesondere das Sprint Planning, die Daily Meetings, die Entwicklungsarbeit, das Sprint Review und die Sprint Retrospektive.

Das **Sprint Planning** ist ein zeitlich begrenztes Ereignis von in der Regel 8 Stunden oder weniger [Time Box], um einen Sprint zu starten. Es dient dem Agilen Team dazu, die Arbeit aus dem Product Backlog, die als nächstes erledigt werden muss, zu evaluieren, zu planen und in den Sprint Backlog zu überführen.

Die **Sprint Retrospective** ist ein zeitlich begrenztes Ereignis von in der Regel 3 Stunden oder weniger [Time Box], um einen Sprint zu beenden. Es dient dem Agile Team dazu, den vergangenen Sprint zu überprüfen und Verbesserungen für den nächsten Sprint zu planen.

Das **Sprint Review** ist ein zeitlich begrenztes Ereignis von in der Regel 4 Stunden oder weniger [Time Box], um die Entwicklungsarbeit eines Sprints abzuschließen. Es dient dem Agilen Team und den Stakeholdern dazu, das aus dem Sprint resultierende Inkrement zu inspizieren, die Auswirkungen der durchgeführten Arbeiten auf den Gesamtfortschritt zu bewerten und die



erfolgreiche Implementierung der betreffenden Anforderungen zu verifizieren. Bei Bedarf wird das Product Backlog auf Basis der Erkenntnisse des Sprint Review angepasst.

Das **Product Backlog** ist eine geordnete Liste von allem, von dem bekannt ist, dass es im Produkt enthalten sein soll. Es dient als einzige Anforderungsquelle für alle Änderungen am Produkt. Der Product Owner ist für das Product Backlog, seine Inhalte, den Zugriff darauf und die Reihenfolge der Einträge verantwortlich. Ein Product Backlog ist niemals vollständig. Während seiner ersten Entwicklungsschritte zeigt es die anfangs bekannten und am besten verstandenen Anforderungen auf. Das Product Backlog entwickelt sich mit dem Produkt und dessen Einsatz weiter. Es ist dynamisch; es passt sich konstant an, um für das Produkt klar herauszustellen, was es braucht, um seiner Aufgabe angemessen zu sein, im Wettbewerb zu bestehen und den erforderlichen Nutzen zu bieten. Die Einträge im Product Backlog sind in Form einer User Story festgehalten.

Das **Product Backlog Refinement** ist ein kontinuierlicher Prozess, in dem der Product Owner und das Entwicklungsteam gemeinsam die Product Backlog- Einträge detaillieren, begutachten, schätzen, priorisieren und revidieren. Der Product Owner kann auch außerhalb des Refinement jederzeit die Einträge im Product Backlog aktualisieren oder aktualisieren lassen.

Das **Sprint Backlog** ist die Menge der für den Sprint ausgewählten Product Backlog- Einträge, die notwendig sind, um das Inkrement zu liefern und das Sprint-Ziel zu erreichen [Tasks]. Das Sprint Backlog ist eine Prognose des Entwicklungsteams darüber, welche Funktionalität im nächsten Inkrement enthalten sein wird, sowie über die erforderliche Arbeit, um diese Funktionalität in einem fertigen [„Done“] Inkrement zu liefern. Das Sprint Backlog macht die gesamte Arbeit sichtbar, die das Entwicklungsteam für notwendig erachtet, um das Sprint- Ziel zu erreichen.

Das **Inkrement** ist das Ergebnis aus allen in

einem Sprint fertiggestellten Product Backlog- Einträgen und ergänzt bzw. erweitert damit das Resultat der Inkremente aller früheren Sprints. Am Ende eines Sprints muss das neue Inkrement fertig [„Done“] sein; das heißt es muss in einem inspizierbaren und verwendbaren Zustand sein. Das Inkrement ist ein Schritt in Richtung einer Vision oder eines Ziels. Es muss auch dann im einsatzfähigen Zustand sein, wenn der Product Owner es aktuell noch nicht ausliefern will. Die Summe aller Inkremente bildet letztlich das entwickelte Produkt.

Impediments sind Hindernisse für das Agile Team, die während eines Sprints auftreten und das Team bei der Erfüllung seiner Aufgabe behindern. Impediments werden durch den Agile Master beseitigt.

Eine **User Story** („Anwendererzählung“) ist eine in Alltagssprache formulierte Anforderung. Sie ist bewusst kurz formuliert und besteht in der Regel nur aus wenigen Sätzen, umfasst dabei das gewünschte Ziel inklusive bestimmter Akzeptanzkriterien. Der Product Owner verfasst die User Stories und fügt diese dem Product Backlog hinzu.

2. Gemeinsames Verständnis zur Abwicklung der Einzelaufträge / Grundregeln

2.1. Selbständige Leistungserbringung / Unterauftragnehmer

2.1.1 Der Auftragnehmer erbringt die vertragsgegenständlichen Leistungen selbständig sowie eigenverantwortlich.

2.1.2 Der Auftragnehmer ist bei der Erbringung seiner Leistungen grundsätzlich in der Wahl des Leistungsorts frei. Erfordert das Projekt jedoch, die Leistungen teilweise in den Räumlichkeiten des Auftraggebers oder eines Dritten durchzuführen, so ist der Auftragnehmer bereit, die Leistungen insoweit in den betreffenden Räumlichkeiten zu erbringen; über den jeweiligen Leistungsort werden sich die Vertragsparteien unter Berücksichtigung der Erfordernisse des Projekts abstimmen.

2.1.3 Der Auftragnehmer hat die alleinige Weisungsbefugnis für die von ihm eingesetzten eigenen Angestellten und etwaige von ihm eingesetzte Unterauftragnehmer. Er ist in der Organisation der Leistungserbringung und in der



Einteilung der Zeit seiner Tätigkeit frei. Er wird sich jedoch insoweit, als das Projekt dies erforderlich macht, mit anderen am Projekt Beteiligten abstimmen zwecks Einhaltung von vereinbarten Terminen.

2.1.4 Im Falle des Einsatzes von Mitarbeitern, Erfüllungsgehilfen und Unterauftragnehmern sichert der Auftragnehmer zu, dass alle erforderlichen behördlichen Genehmigungen (wie z. B. Arbeitsgenehmigung, Aufenthaltstitel) vorliegen. Der Auftragnehmer stellt den Auftraggeber von sämtlichen Rechtsfolgen frei, die sich aus der Nichteinhaltung dieser Anforderung ergeben.

2.1.5 Der Auftragnehmer und dessen Unterauftragnehmer sind für die Erfüllung von sämtlichen Arbeitgeberpflichten verantwortlich, insbesondere auch in Bezug auf die Zahlung von Gehältern und Sozialversicherungsbeiträgen. Der Auftraggeber haftet nicht für die Zahlung - insbesondere - von Gehältern, Tagessätzen, Steuern, Sozialversicherungsbeiträgen und Versicherungsbeiträgen für vom Auftragnehmer bzw. dessen Unterauftragnehmern eingesetzte Kräfte. Weder ein Rahmenvertrag noch ein Einzelauftrag begründen einen Arbeitsvertrag zwischen dem Auftraggeber und irgendeiner bei dem Auftragnehmer oder einem Unterauftragnehmer beschäftigten Person.

2.1.6 Die Beauftragung eines Unterauftragnehmers (hierzu zählen auch externe Berater und Freiberufler) bedarf der schriftlichen Zustimmung des Auftraggebers, welche ohne die Angabe von Gründen versagt werden kann. Vom Auftragnehmer gewünschte Unterauftragnehmer müssen im Angebot mit Namen des Beraters und den Firmendaten ausgewiesen werden. Verbundene Unternehmen des Auftragnehmers sind ebenfalls Unterauftragnehmer im Sinne dieser Ziffer.

2.2. Mitwirkung A1/ Agile Teams in der Projektorganisation

Nach der agilen Softwareentwicklungsmethode arbeiten die Mitarbeiter des

Auftraggebers und des Auftragnehmers in AGILE-Teams zusammen. Üblicherweise besteht die Projektorganisation aus einem Product Owner (vom Auftraggeber gestellt), einem Agile-Master (wird normalerweise vom Auftraggeber gestellt), und dem Entwickler-Team (wird vom Auftragnehmer gestellt, kann teilweise auch Mitarbeiter des Auftraggebers, bzw. anderer Vendorn enthalten – „Mixed Teams“). Somit ist die Mitwirkung der A1 überall dort notwendig, wo A1-MitarbeiterInnen Rollen und Verantwortlichkeiten im Zusammenhang mit dem Agile Team haben.

Dazu gehören:

- Planung aller Termine und Sprints zusammen mit dem AN
- Priorisierung der Anforderungen und Erstellung des Backlogs für die Sprints
- Abhaltung regelmäßiger Daily Meetings, Sprint Meetings und Sprint Reviews

Im Rahmen der Erbringung von Leistungen aus diesem Leistungsbaustein sind durch A1 als Auftraggeber folgende Mitwirkungspflichten zu erfüllen:

- Spezifizierung der Anforderungen für die zu erstellende Software, und Übermittlung dieser in Form von klar definierten Grundfunktionalitäten an den Auftragnehmer (Requirements)
- Nominierung von Mitarbeitern als Ansprechpartner für Abstimmungen, reguläre Projektbesprechungen, Sprints, Eskalationen etc.
- Einweisung in Entwicklungs- und Inbetriebnahmeprozesse des Auftraggebers
- Bereitstellung relevanter Dokumentationen, Richtlinien und Guidelines zu Prozessen, Architekturen und Applikationen
- Aufbereitung und rechtzeitige Bereitstellung von Informationen, Daten, Hardware, Betriebssystemen, eingesetzter Standardsoftware, Organisationspläne und sonstiger für die Programmherstellung nötiger Komponenten
- Abnahme und Prüfung der einzelnen Deliverables wie Konzepte, Basis-Software, Endsoftware, Protokolle etc. und zügiges Feedback an Auftragnehmer (schriftlich).



Störungen und Impediments müssen vom Auftragnehmer bzw. den von ihm eingesetzten Personen über den Agile Master kanalisiert werden und dürfen nicht eigenständig mit anderen Mitarbeitern in der Organisation des Auftraggebers geklärt werden. Auch sonstige Rückfragen des Auftragnehmers bzw. der von ihm eingesetzten Personen, z.B. an den Product Owner während eines Sprints, müssen über den Agile Master koordiniert werden.

3. ERGEBNISSE (DELIVERABLES)

Durch diesen Leistungsbaustein werden typischerweise folgende Ergebnisse geliefert:

- Softwarekonzept- und Design
- Individuell angefertigte, voll funktionsfähige Software und Customizing (Anpassungen, Erweiterungen)
- Source Code und eine diesen beschreibende und erläuternde Dokumentation, deren Mindestumfang so zu bemessen ist, dass nach angemessener Einarbeitungszeit ein Verständnis des Aufbaus und der Arbeitsweise des Programms ermöglicht wird
- Dokumentation der gelieferten Softwarefunktionalität (Betriebsführungshandbuch, User Manual, Installationsanleitung)
- Schulungen, Anwendertraining

4. ABNAHME

Nach jedem Sprint erfolgt die Abnahme der Deliverables (*User Stories*), die in diesem Sprint umgesetzt wurden. Der AN erklärt gegenüber AG, dass die für den Sprint gemeinsam vereinbarten User Stories fertig sind, und den zu Beginn des Sprints vereinbarten Abnahmekriterien entsprechen (*Definition of Done*). Grundsätzlich wird ein Sprint nur in seiner Gesamtheit abgenommen – d.h. wenn alle vereinbarten User Stories allen

vereinbarten Kriterien entsprechen. Die Sprint-Termine werden zwischen Product Owner, Agile Master und Team vereinbart, und können schon im Angebot angeführt werden.

Die finale Abnahme des Gesamtprodukts (*Final Acceptance*) erfolgt nach Fertigstellung der finalen Version der voll funktionsfähigen, für den Rollout bereiten Software, bzw. nach der Installation des Programms auf der Hardware des Auftraggebers und der Ersteinweisung.

Die Abnahme erfolgt mindestens in folgenden Schritten:

- nach der Installation des Programms weist der Auftragnehmer durch angemessene Abnahmetests das Vorhandensein der vereinbarten Eigenschaften und der wesentlichen Funktionalitäten und
- nach erfolgreichen Abnahmetests erstellen Auftragnehmer und Auftraggeber gemeinsam ein Abnahmeprotokoll, in dem auch evtl. Mängel festzuhalten sind und
- vom A1-Verantwortlichen wird eine schriftliche Abnahmeerklärung abgegeben.



5. KEY PERFORMANCE INDICATORS (KPIs)

Während bei der Wasserfallmethode die KPI erst nach Abschluss eines Projekts eine quantitative Wertung über Projekterfolg- und Effizienz erlauben, sind bei agiler Softwareentwicklung zwei Kategorien von KPI aussagekräftig. Neben den KPI die nach Abschluss des Projekts ermittelt werden, gibt es in der agilen Softwareentwicklung auch aufschlussreiche KPI die bereits während der Projektlaufzeit Auskunft über den aktuellen Status eines agilen Projekts bzw. den Projektfortschritt geben.

5.1. Zielwerte und Messung während der Projektlaufzeit

5.1.1. Sprint Burndown (Projektfortschritt; Produktivität)

Definition und Zielwert	<ul style="list-style-type: none">• Story Points/Sprint; wird individuell für jedes Projekt vereinbart• Sprint Burndown ist der Grad des Abbau bzw. der Fertigstellung der Backlog-Tasks in einem laufenden Projekt. Er zeigt somit den aktuellen Fortschritt eines Projekts.• Sprint Burndown hängt direkt mit der Velocity zusammen (siehe 3.1.2) und wird – so wie diese – in Story Points/Sprint gemessen.
Messmethode	Basis für die Messung ist der Sprint Burndown Chart (Fertigstellungschart). Er zeigt den Task-Abbau im Zeitablauf.

5.1.2. Team Velocity (Projektfortschritt; Produktivität)

Zielwert	<ul style="list-style-type: none">• Story Points/Sprint; wird für jedes Projekt individuell vereinbart.• Bei neuen (noch nicht eingespielten) Teams/Methoden sollte auch vereinbart werden, um welchen Wert die Velocity im Durchrechnungszeitraum steigen sollte. <p>Die Velocity soll nämlich mit der Zeit steigen, da sich das AGILE-Team irgendwann gut eingespielt und Know-How und Erfahrung gesammelt hat.</p>
Messmethode	Anzahl der Story Points, die ein Team in einem Sprint umgesetzt hat. Basis für die Messung sind der Sprint Burndown Chart und die Daten aus den A1 Tracking Systemen.



5.1.3. Scope Creep (In-Budget Delivery)

Definition	<ul style="list-style-type: none"> • User Stories/Sprint • Anzahl der zusätzlichen User Stories/Sprint die dem Originalscope nachträglich hinzugefügt wurden. Scope Creep entspricht somit einer Scope-Erweiterung während der Laufzeit des Projekts.
Messmethode	Differenz zwischen der tatsächlichen Anzahl der User Stories im Sprint und der ursprünglich im Scope geplanten User Stories

5.1.4. Qualität

Definition	<ul style="list-style-type: none"> • Defects/User Story • Max. Anzahl der Defects wird individuell vereinbart pro Stage (INT/PROD,..)
Messmethode	Anzahl bestehender Defects/User Story pro Stage. Kategorisierung der Defects folgt der Klassifizierung und Auswertung im Tracking System der A1.

5.1.5. Automatisierte Test-Abdeckung (Qualität)

Definition und Zielwert	<ul style="list-style-type: none"> • Anteil des durchlaufenen Codes am gesamten Programmcode beim Ausführen von Tests • Zielwert: min. 80%
Messmethode	<ul style="list-style-type: none"> • Vollautomatisierte Tests • Berechnung (%) = (Anzahl durchgelaufener Code-Zeilen/Gesamtanzahl der Code-Zeilen)*100

5.2. Zielwerte und Messung nach Projektabschluss

5.2.1. In-Budget Delivery

Zielwert	€ 0,-- Budgetüberschreitung
Messmethode	Umgesetzte Deliverables im Verhältnis zum (noch) verfügbaren geplantem Budget

5.2.2. Unit Cost

Zielwert	EUR-Wert fix vorgegeben pro Projekt
Messmethode	EUR/Story Point



5.2.3. Delivered Defect Density to UAT

Zielwert	Maximalwert individuell vereinbart
Messmethode	Anzahl der Fehler pro Epic/EUR/FP/Jira req.

5.2.4. Resolution Time for Blocker

Zielwert	≥90% aller Blocker in UAT innerhalb von 4 Std. gelöst
Messmethode	Anzahl der Blocker die pro Zeiteinheit gelöst werden

5.2.5. Iterations to solve INT defects

Zielwert	>90% werden in der ersten Iteration gelöst.
Messmethode	Anzahl der Versuche die zum Lösen eines Tickets führen