

Entwicklungsphase	Herkömmliche Produktentwicklung	Komponenten-basierte Produktentwicklung (ACF)
Entstehung des Bedarfs an einem Produkt auf dem Markt		<p>Vervollständigung bzw. Erweiterung der vorhandenen Komponentenpalette</p> <p>Bessere Überbrückung von Projektleerlaufzeiten</p> <p>Generierung weiterer Produktideen</p> <p>Schneller Marktzugang durch effizientes Prototyping</p>
Systemanforderungsanalyse	<p>Auswahl der Technologie (vorhanden oder nicht)</p> <p>Eventuell Erweiterung vorhandenen Produkte</p> <p>Erstellung einer Anforderungsspezifikation</p>	<p>Basistechnologie ist festgelegt</p> <p>Nur aufgaben-spezifischen Systemkomponenten müssen definiert werden</p> <p>Alle entwickelten Komponenten stehen zur Verfügung</p> <p>Erstellung einer Anforderungsspezifikation mit dem Ziel, die noch fehlenden Komponenten, zu definieren</p>
Systemarchitektur	<p>Festlegung der Anwendungsfälle</p> <p>Definition der Systemkomponenten</p> <p>Festlegung der Systemschnittstellen</p>	<p>Festlegung der Anwendungsfälle unter Berücksichtigung existierender und sich bewährter Templates</p> <p>Betrachtung der Übertragungsmöglichkeit der neu einzuführenden Templates auf die zukünftige Produkte</p> <p>Auf die neuen Komponenten reduzierte Festlegung der Systemschnittstellen</p>
Systementwurf	<p>Machbarkeitsstudien</p> <p>Abschließende Festlegung der Systemkomponenten</p>	<p>Machbarkeitsstudien für fehlende Konzepte unterstützt durch produktnahes Prototyping</p> <p>Abschließende Festlegung der Systemkomponenten</p>
Softwarearchitektur	<p>Entwurf einer detaillierten Strategie zu Realisierung der definierten Anforderungen</p> <p>Entwurf eines ersten Prototyps</p>	<p>Inter-Kommunikation zwischen den Softwaremodulen wird durch das Framework verwaltet</p> <p>Während der gesamten Entwicklungsphase, ist es möglich, die notwendigen Prototypen sehr schnell zu erzeugen</p> <p>Einzelne Funktionsteile der Anwendung können unabhängig voneinander erzeugt und getestet werden</p>

Entwicklungsphase	Herkömmliche Produktentwicklung	Komponenten-basierte Produktentwicklung (ACF)
		Framework motiviert zu einer strikten Trennung der Funktionalitäten durch die wohldefinierten Schnittstellen. Dadurch können Fehlentscheidungen in der Frühphase des Projekts vermieden werden
Softwarerealisierung	Umsetzung der festgelegten Ziele	Umsetzung der fehlenden Produktbausteinen Verteilte Softwareentwicklung möglich
Softwaretests	Verifikation der Ergebnisse der Softwarerealisierung	Verifikation der Ergebnisse der Softwarerealisierung ist partiell möglich Testiterationen können kürzer gehalten werden Tests können beliebig skaliert werden Template-basierte Tests beschleunigen den gesamten Testzyklus
Integrationstests	Verifikation der Softwarearchitektur	Partielle Verifikation ermöglicht frühere Erkennung der Schwachstellen der Systemarchitektur
Systemintegration	Verifikation des Gesamtsystems	Verifikation des Gesamtsystems Reaktionszeit auf mögliche Probleme ist viel kürzer, da das Fehlverhalten im Normalfall dank der Trennung auf geschlossene Bausteine nur auf wenige Komponenten begrenzt ist Durch die breite Basis an bereits getesteten Komponenten reduziert sich die Wahrscheinlichkeit der System-Reintegration

Entwicklungsphase	Herkömmliche Produktentwicklung	Komponenten-basierte Produktentwicklung (ACF)
Abnahme und Nutzung	<p>Aus Kundensicht</p> <ul style="list-style-type: none"> • Produkte desselben Unternehmens können sich sehr stark voneinander unterscheiden • Standardisierte Benutzerschnittstelle schwierig umsetzbar • Nachträgliche Produkterweiterung zu teuer • Standardumfang des Produkts ist reduziert auf vereinbarte Funktionalität <p>Aus Unternehmenssicht</p> <ul style="list-style-type: none"> • Dieselben Bugs müssen in jedem Produkt separat gepflegt werden • Modulare Feature-Erweiterungen nur mit hohem Aufwand möglich • Produkterweiterungen nicht verteilbar zwischen unterschiedlichen Produkten • Bei negativer Projektentwicklung (bspw. Projektabbruch) geht der geleistete Entwicklungsaufwand verloren 	<p>Aus Kundensicht</p> <ul style="list-style-type: none"> • Produkte desselben Unternehmens haben die gleiche Anwendungslogik und Erscheinungsbild. • Identifikation mit neuen Produkten dadurch viel leichter • Ein geringerer Einarbeitungsaufwand notwendig • Standardisierte Benutzerschnittstelle • Nachträgliche Produkterweiterungen können schnell und kostengünstig umgesetzt werden • Standardumfang des Produkts muss nicht auf die vereinbarte Funktionalität reduziert werden und kann von anderen Entwicklungen profitieren <p>Aus Unternehmenssicht</p> <ul style="list-style-type: none"> • Fehler müssen nur einmal für alle betroffenen Produkte behoben werden • Modulare Feature-Erweiterungen sind mit einem geringerem Aufwand möglich • Produkterweiterungen verteilbar zwischen unterschiedlichen Produkten • Nutzung in weiteren Projekten/Produkten (unabhängig vom Erfolg des eigentlichen Vorhabens)