



# White Paper

# Kontinuierliche

# Verbesserung von

# Software



[www.red6-es.de](http://www.red6-es.de)

# Kontinuierliche Verbesserung von Software

## Ein systematisches Vorgehen nach der Architecture Improvement Method aim42

Ein Continuous Improvement Process (CIP) wie z.B. Kaizen wird oft für eine Organisationseinheit, ein Prozess oder begleitend für ein Projekt etabliert.

Wie wäre es aber, ein CIP auf den Lifecycle einer Software auszurichten?

Genau dieser Ansatz wird mit der Architecture Improvement Method aim42<sup>1</sup> verfolgt. Ein Erfahrungsbericht.

von Jan Callesen, Senior Software Architect

## Einleitung und Problemstellung

Software und ihre Umgebungen werden über Monate und Jahre umfangreicher und komplexer. Die Verständlichkeit von Softwaresystemen und Prozessen verringert sich. Neue Funktionalität und Änderungen an Geschäftsprozessen erfordern stetige Anpassungen. Ein hoher Bedarf an digitalisierten Produkten und Dienstleistungen ist eine Chance für neue Geschäftsideen und steigert die Frequenz der Änderungswünsche an existierenden Softwarelösungen zusätzlich. Gleichzeitig sind die Kosten für Änderungen zu hoch oder riskant und resultieren in zu vielen Fehlern.

---

<sup>1</sup> aim42 ist eine systematische Vorgehensweise zur Verbesserung von Softwaresystemen und -architekturen. aim42 wurde 2013 von Gernot Starke gegründet und seit 2014 zusammen mit Stefan Tilkov weiterentwickelt. Die Informationen über aim42 (<https://www.aim42.org>) werden in diesem White Paper gemäß der *Creative Commons ShareAlike* Lizenz veröffentlicht.

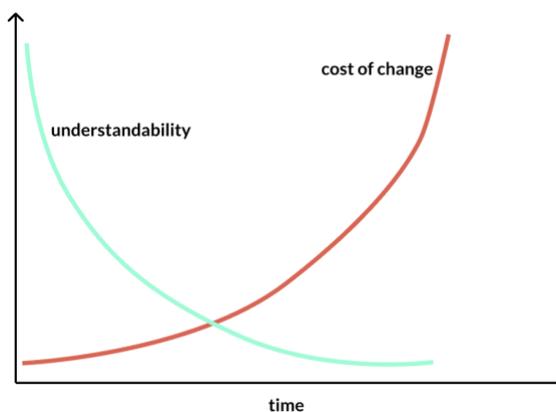


Abbildung 1, aim42, www.aim42.org

Eine mangelhafte Qualität, beispielsweise in den Bereichen Performance und Security sind weitere Gründe, die für eine Investition in die Verbesserung bestehender Software-systeme sprechen.

Ein Vorgehen zur Verbesserung ist die frei verfügbare Architecture Improvement Method aim42. Neben einem schlanken Vorgehensmodell umfasst sie einen großen Katalog an Praktiken zur evolutionären und systematischen Modernisierung sowie Verbesserung von Softwaresystemen. Die damit verfolgten langfristigen Ziele sind vor allen anderen eine gute Verständlichkeit und damit einhergehenden linearen Kosten für Softwareänderungen.

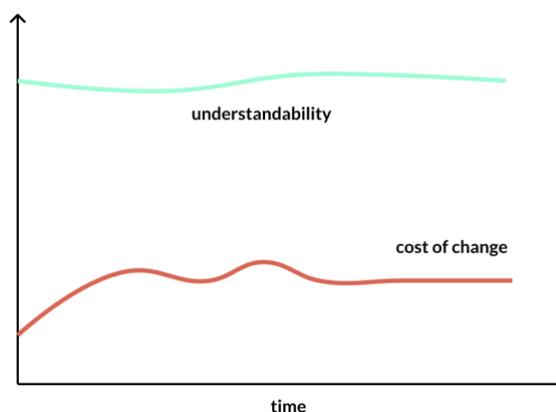


Abbildung 2, aim42, www.aim42.org

## Das systematische Vorgehen aim42

Vergleichbar mit Scrum geht aim42 iterativ vor, unterteilt in drei Phasen: Analyze, Evaluate, und Improve. Für jede Phase schlägt aim42 aus seinem Katalog etablierte Praktiken vor, beschreibt diese oder referenziert auf weiterführende Informationen. Zur Unterstützung und Steuerung dieses Vorgehens gibt es übergreifende (Crosscutting) Aktivitäten, die in allen drei Phasen zur Geltung kommen.

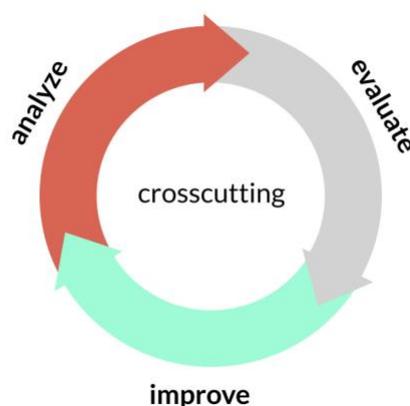


Abbildung 3, aim42, [www.aim42.org](http://www.aim42.org)

Eine wichtige Aktivität ist das Schreiben und Aufnehmen von Issues und möglichen Improvements. Beispielsweise mithilfe eines Ticket-Systems wie JIRA. Bei der Aufnahme neuer Tickets gibt es zwei Grundregeln:

1. Sowohl Issues als auch Improvements sollen in einer Business-Einheit bewertet werden wie beispielsweise Zeit oder idealerweise Kosten in Euro. Dies ermöglicht eine Objektivierung der Prioritäten und hilft diese gegenüber Stakeholder zu verargumentieren.
2. Issues und Improvements müssen aufeinander bezogen sein. Zu einem Improvement muss es ein oder mehrere Issues geben. Anders formuliert: „Keine Problemlösung ohne ein Problem.“

## Die Analyze-Phase

Die Ziele in dieser Phase sind vor allem das Finden von Problemen, Defiziten wie z. B. hohe Aufwände bei Softwareänderungen, Fehlern, regelmäßigem Ärger, technische Schulden, usw. Die Suche bezieht sich nicht nur auf die Softwaresysteme selbst, sondern auch auf den Betrieb der Software, der Wartung, den Entwicklungsprozess, fachliche Prozesse. In diesem

Prozessschritt werden Issues und mögliche Improvements explizit voneinander getrennt. Dies verhindert u. a. zu früh in Lösungen zu denken, ohne sich hinreichend mit dem eigentlichen Problem auseinandergesetzt zu haben. Das Verständnis von internen Strukturen, fachlichen und technischen Konzepten sowie von architektonischen Ansätzen zu erhöhen und zu dokumentieren ist ein weiteres Ziel in dieser Phase. Diese Ziele sind autonom zu erreichen. Sie erfordern ein gutes Feedback und eine zeitweise enge Zusammenarbeit mit Knowhow-Trägern und Stakeholder. Für die Zusammenarbeit mit diesen Experten beschreibt die aim42 Methode einen großen Katalog an erprobten Aktivitäten, u. a.:

- Stakeholder Analysis
- Kontext Analysis
- Root Cause Analysis

Die Beschreibungen dazu sowie viele weitere mögliche Aktivitäten, Maßnahmen und Referenzen sind unter folgender URL in englischer Sprache einsehbar: [https://aim42.github.io/#\\_patterns\\_and\\_practices\\_for\\_analysis](https://aim42.github.io/#_patterns_and_practices_for_analysis)

## Die Evaluate-Phase

In dieser Phase werden die gesammelten Issues mit den dazugehörigen möglichen Improvements bewertet. Das Ziel ist es diese zu objektivieren sowie gegeneinander abzuwägen. Issues und Improvements mit dem besseren Verhältnis zwischen den voraussichtlichen Kosten und Nutzen erhalten eine höhere Priorität für die Umsetzung konkreter Verbesserungsmaßnahmen.

Die Bewertung von Issues und Improvements basiert zum einen auf Messungen. Beispielsweise kann die Zeit gemessen werden, die Nutzer für etwaige Workarounds benötigen, um Geschäftsprozesse mit einer ineffizienten oder fehlerbehafteten Software zu bearbeiten. Zum anderen kann der Wert von Issues und den dazugehörigen Improvements geschätzt werden. Bei diesen Schätzungen sollte nicht von absoluten Werten ausgegangen werden. Realistischer sind Intervalle wie zum Beispiel die möglichen minimalen und maximalen Kosten, nach Möglichkeit unter der Angabe von Wahrscheinlichkeiten.

Einzelne Kostenfaktoren sind nicht in jedem Fall mess- oder schätzbar. In diesen Fällen können Annahmen getroffen werden. Explizite und offen gelegte Annahmen helfen, bessere Entscheidungen bei der Priorisierung zu treffen. Wenn beispielhaft der Support-Vertrag von einem Dienstleister einer Drittsoftware ausläuft, ist anzunehmen, dass Änderungen an dieser Software ohne Support aufwändiger sind.

Die Referenzen der Methoden zur Evaluation finden sich unter der URL: <https://aim42.github.io/#Evaluate>

## Die Improve-Phase

In der dritten Phase, Improve werden Verbesserungsaktivitäten koordiniert und durchgeführt, um die dazugehörigen Issues zu beseitigen, die vor allem während der Phase Analyse festgestellt wurden. Die möglichen Aktivitäten können u. a. sein:

- Code, Strukturen, Konzepte oder Prozesse ändern, um bessere Software zu erreichen
- Reduzierung von technischen Schulden
- Optimierung von qualitativen Attributen wie bspw. Performance, Security oder Wartbarkeit
- Optimierung von Betriebs- und Verwaltungsprozessen und damit von Aufwand und Kosten

Die Phase ist wie in Abbildung 4 strukturiert.

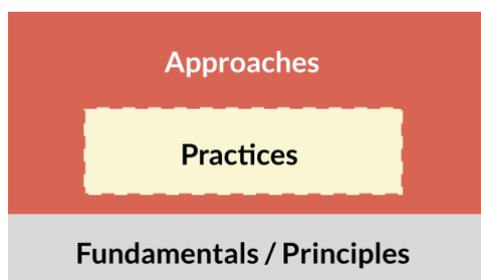


Abbildung 4, aim42, [www.aim42.org](http://www.aim42.org)

### Grundlagen

Die grundlegenden Prinzipien, die bei allen Schritten auf dem Weg zur Verbesserung berücksichtigt werden sollten. Unabhängig von der aim42 Methode helfen diese jedem professionellen Software-Entwicklungsteam.

### Herangehensweisen

Allgemeine strategische sowie langfristige Entscheidungen, wie Verbesserungen angegangen werden können, strukturiert in mehrere Kategorien (siehe Abbildung 5).

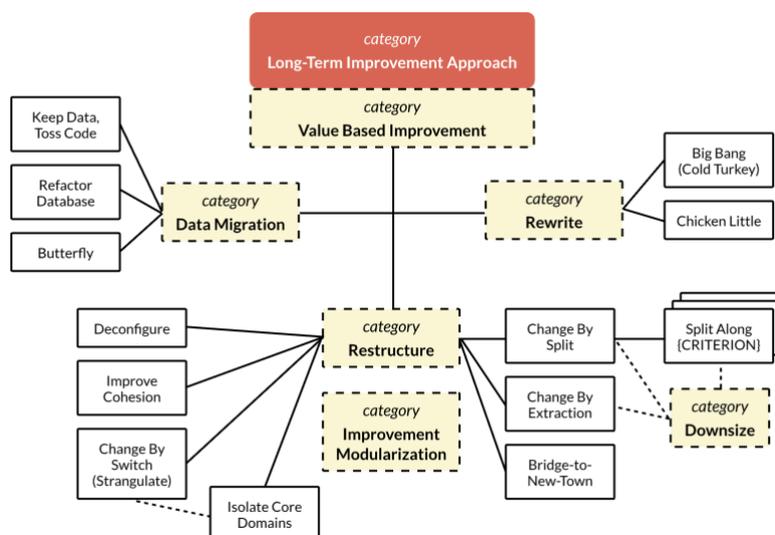


Abbildung 5, aim42, [www.aim42.org](http://www.aim42.org)

## Praktiken

Kleinteilige Praktiken und Muster, die in mehrere Kategorien unterteilt sind. Diese sind der Großteil der aim42 Methodenreferenz, sie sind unter folgender URL beschrieben: <https://aim42.github.io/#improve-approaches-details>

## Implementierung und Integration im Tagesgeschäft

Die Einführung der aim42 Methode ist vergleichbar mit der Einführung anderer Methoden in der Softwareentwicklung, wie zum Beispiel Scrum. Wichtig ist ein gemeinsames Verständnis der Methode und der dazugehörigen Aktivitäten innerhalb des Teams, welches aim42 in seine Arbeitsorganisation integriert. Für ein Verständnis der Grundlagen kann dieses Whitepaper dienen.

Analog zu Scrum ist aim42 iterativ. Dies fördert das Lernen der drei Phasen Analyse, Evaluate und Improve durch Wiederholung und Reflexion.

Um das Team mit dieser neuen Methode nicht zu überfordern, bietet es sich an, bei der Implementierung selbst iterativ vorzugehen: Anstatt sich mit allen Phasen gleichermaßen auseinanderzusetzen, kann sich das Team in einer begrenzten Zeit ausschließlich mit Aktivitäten der Phase Analyse beschäftigen. Wie im Kapitel „Die Analyse-Phase“ beschrieben, gehört dazu das Kennenlernen und Dokumentation des Systems und der Prozesse sowie die Analyse und das Auffinden von Issues. Auch innerhalb der Phase ist es

effektiver, wenige der Praktiken, die aim42 in dieser Phase empfiehlt, anzuwenden, anstatt zu viele gleichzeitig. Letzteres birgt das Risiko, sich zu verzetteln und das Team zu überfordern. Hier gilt: Qualität über Quantität.

Eine gute Wahl zweier erprobter Praktiken aus der Phase Analyse sind zum einen die Stakeholder Analysis und zum anderen die Root Cause Analysis.

Bei der vorgeschlagenen Einführung von aim42 in mehreren Schritten, folgt das Lernen und Anwenden der Praktiken der Phase Evaluate nach der Phase Analyze und im Anschluss danach in dritten Phase Improve.

Zur Unterstützung des Teams beim Lernen und bei der Implementierung der aim42 Methode empfiehlt sich eine dedizierte Rolle innerhalb des Teams. Diese Rolle befähigt das Team, die vorgeschlagenen Praktiken aus allen Phasen von aim42 umzusetzen und sorgt zusammen zum Beispiel mit der Projektleitung oder dem Product Owner für einen geeigneten Rahmen. Eine gute Analogie dieser Rolle ist der Scrum Master bei der Einführung der Methode Scrum. Für die Implementierung und Integration von aim42 werden Ressourcen benötigt. Es erfordert eine stetige Investition in die Verbesserung von Software und den dazugehörigen Prozessen. Ein geeigneter Rahmen beginnt mit der Budgetplanung. Es sollte vorher klar sein, wie viel in die Aktivitäten zur Verbesserung investiert werden soll. Das langfristige Ziel dabei ist ein gleichbleibendes Niveau der Kosten für Änderungen an einer Software in ihrem gesamten Lebenszyklus (siehe Abbildung 2). Wenn das Tagesgeschäft beispielhaft vor allem aus der Fehlerbereinigung und dem Betrieb der Software sowie der Implementierung neuer Funktionalität besteht, kann eine Verteilung des Budgets und der Ressourcen wie im folgenden Diagramm aussehen:

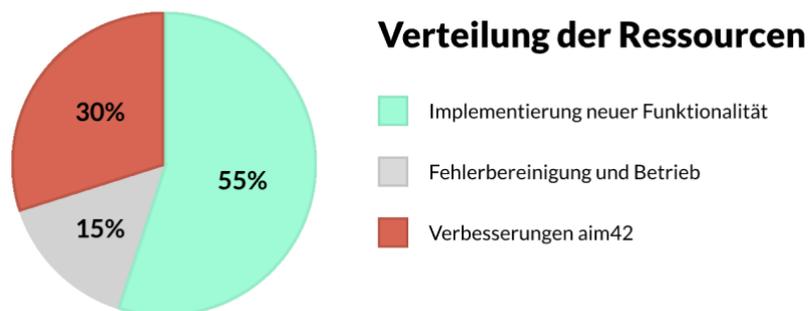


Abbildung 6, aim42, [www.aim42.org](http://www.aim42.org)

Im Tagesgeschäft kann aim42 parallel zu dem bestehenden Vorgehen implementiert werden. Die bisherige Arbeitsorganisation muss nicht geändert werden. Wird zum Beispiel mit der Scrum-Methode gearbeitet, kann aim42 parallel dazu wie in Abbildung 7 dargestellt verstanden werden.

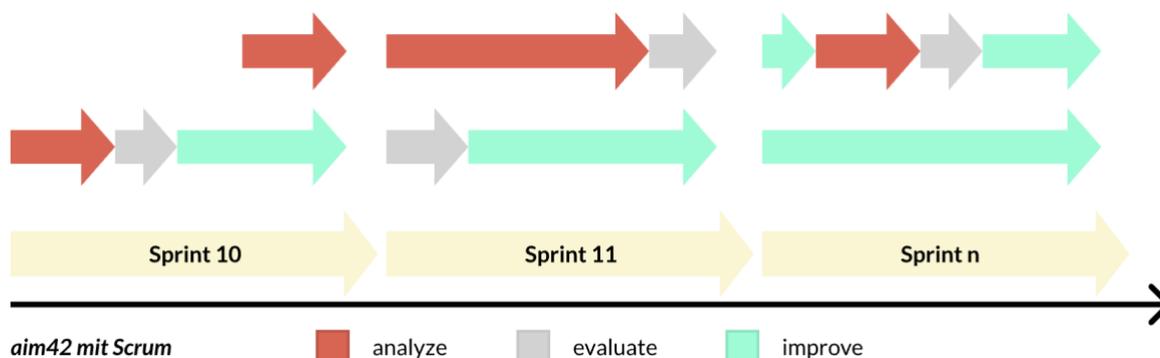


Abbildung 7, aim42 zusammen mit Scrum

## So gelingt eine reibungslose Adoption von aim42: 3 Tipps aus der Praxis

### 1. Sorgen Sie zunächst für den Wissensaufbau

Damit die Einführung von aim42 im Entwicklungsprojekt oder im Betrieb Ihrer Software möglichst reibungslos gelingt empfehlen wir, mit dem Wissensaufbau im Team anzufangen. Eine möglichst breite Basis an Mitarbeitenden sollte Zeit und Raum erhalten, die Methode zu verstehen und sich mit ihren Fragestellungen im Team auseinander zu setzen. Wenn alle im Team dasselbe Zielbild von der aim42 Methode und die Phasen haben, wird die Einführung der Methode und der Übergang in die Routine insgesamt effizienter für alle.

### 2. Beginnen Sie mit einer weniger komplexen Software

Damit Ihr Team sich möglichst frühzeitig von der Wirkung der aim42 Methode selbst überzeugen kann lohnt es sich, nicht mit der umfangreichsten und komplexesten Software zu starten, sondern mit einer kleineren und weniger komplexen Anwendung. Diese Vorgehensweise erlaubt ihrem Team, schneller wertvolle Erkenntnisse in den Evaluate- und Analyze-Phasen zu gewinnen und als Team erste Erfolge in der Umsetzung von Verbesserungen zu realisieren. Dadurch erhält Ihr Team einen wichtigen Motivationsschub, der gerade zu Beginn der Einführung wichtig ist.

### 3. Bleiben Sie den Grundregeln treu

Das regelmäßige Abstimmen von Prioritäten mit Stakeholdern auf der Grundlage Ihrer Bewertung von Issues und Improvements in den Aufwandskategorien Zeit und Kosten in Euro baut Vertrauen auf. Bleiben Sie also beharrlich bei der quantitativen Bewertung von Issues und Improvements. Erlauben Sie Ihr Team nicht an Problemlösungen zu arbeiten, für die es kein Problem gibt. Ein Rückfall in alte Gewohnheiten kann schnell passieren!

## Fazit

aim42 ist eine praxisorientierte und effektive Methode zur kontinuierlichen Verbesserung von Softwaresystemen. Durch die Kombination der besten Praktiken aus der Softwareentwicklung, agilem Management sowie leichtgewichtigen Prinzipien erreicht aim42, dass Teams hochqualitative Softwareprodukte schnell und effizient produzieren können. Wir empfehlen die aim42 Methode als systematischen Ansatz zur kontinuierlichen Verbesserung und bieten Unterstützung für die Implementierung in jeder Phase von aim42 in der Praxis.

## Sie möchten mehr erfahren?

Unsere red6 Experten beraten Sie gerne unverbindlich zu passenden Lösungsansätzen.

**Felix Gerdes, Business Development**

[felix.gerdes@red6-es.de](mailto:felix.gerdes@red6-es.de)

[040 4119 2915](tel:04041192915)

[www.red6-es.de](http://www.red6-es.de)

