



## INHALTSVERZEICHNIS

Einführung.....	2
Warum Schnelligkeit und Qualität in der Entwicklung zwei gute Freunde sein können.....	3
10 Tipps wie Schnelligkeit und Qualität in der Praxis möglich gemacht werden können .....	4
Die wichtigsten Erkenntnisse .....	6
Grundlagen Anforderungskatalog.....	6
Ein dynamischer Prozess.....	6
Kurze Feedback Loops.....	6
Best Practices .....	7
Fazit .....	8
Wichtige Schlussfolgerungen .....	9



## SOFTWAREENTWICKLUNG – SCHNELLIGKEIT & QUALITÄT

### Warum Schnelligkeit und Qualität zwei gute Freunde sein können

Der Erfolg moderner Unternehmen hängt heutzutage mehr denn je von ihrem technischen Know-how und ihrer Fähigkeit ab, Innovationen schnell umzusetzen. Die Schnelligkeit der Softwareentwicklung bei der Umsetzung neuer Ideen, Technologien und Prozesse wiederum trägt entscheidend dazu bei, einen Wettbewerbsvorteil gegenüber Mitbewerbern zu erlangen. Doch worin besteht die Schnelligkeit an sich, welche Aspekte müssen bei der Entwicklung berücksichtigt werden und wie kann sie sichergestellt werden? In diesem Whitepaper sollen

deshalb die relevanten Eigenschaften und Prinzipien der schnellen Softwareentwicklung vorgestellt werden.

Schnelligkeit in der Softwareentwicklung ist ein komplexes Phänomen, das mehrere Dimensionen umfasst. Zunächst geht es darum, die Gesamtzeit der Entwicklung eines bestimmten Produktes möglichst kurz zu halten. Dazu werden mobile Iterationen eingesetzt und agile Arbeitsmethoden unterstützt, die es ermöglichen, kontinuierlich neue Funktionen oder Funktionsteile bereitzustellen. Zudem ist es wichtig, den Code möglichst einfach und klar zu strukturieren, um das Debugging und Testens zu erleichtern. Auch bei der Testsimulation sind Ergebnisse umsetzbar, die in nur wenigen Iterationen erzielt werden können.

Zugleich ist die Schnelligkeit in der Softwareentwicklung auch eine Frage des Designs. Für ein gutes Software-Design müssen Datenübermittlung, Fehlerbehandlung und Datenübertragung so unkompliziert wie möglich gestaltet werden. Dazu können Designmustern und -prinzipien, sowie Methoden der Automatisierung und des übergreifenden Testens angewendet werden. Auch muss beachtet werden, dass die Skalierbarkeit gewährleistet ist und die Funktionalität der Software zu jeder Zeit zuverlässig arbeitet.



Schließlich muss die Schnelligkeit in der Softwareentwicklung auch in der Organisation des Teams sichergestellt werden. Hierfür können verschiedene Prozesse implementiert werden, die das Zusammenarbeiten vereinfachen. Dazu gehören beispielsweise das einheitliche und klar gekennzeichnete Code-Review, Agile-Meetings, Echtzeit-Arbeitssynchronisation und Retrospektiven. All diese Prozesse können helfen, das unternehmensübergreifende Zusammenarbeiten zu vereinfachen und sicherzustellen, dass das gesamte Entwicklungsteam auf dem gleichen Stand ist.

Insgesamt kann gesagt werden, dass die Schnelligkeit in der Softwareentwicklung ein wesentlicher Faktor für den Unternehmenserfolg ist. Viele verschiedene Aspekte müssen dabei beachtet werden, darunter Code-Struktur, Design, Testsimulation und Organisation des Entwicklungsteams. Mit diesen Methoden ist es möglich, das Produkt schnell auf den Markt zu bringen und einen Wettbewerbsvorteil zu erzielen.



## 10 TIPPS WIE SCHNELLIGKEIT UND QUALITÄT IN DER SOFTWAREENTWICKLUNG MÖGLICH SIND

1. Einen Kulturwandel herbeiführen: passe das Unternehmensprofil so an, dass es den Fokus auf die Lieferung hochwertiger Software in kürzester Zeit legt. Stelle deinem Team methodisches Wissen und Tools zur Verfügung, um eine zügige und qualitativ hochwertige Arbeit zu ermöglichen.
2. Ein agiles Verständnis vermitteln: Nutze Kalender- Meetings und verabschiede monatliche Berichte zugunsten der regelmäßigen Updates und Begutachtung des aktuellen Fortschritts.
3. Starthilfe geben: Fahre Projekte mit einer Struktur und Erläuterungen an, um Prioritäten und Ziele besser herauszuarbeiten.
4. Kommunizieren: Um ein Gefühl für die Performance und Qualität der Arbeit des Teams zu erhalten, muss eine Kommunikation zwischen Entwicklern und Führungskräften gestärkt werden.
5. Planung: Legen Sie feste Ziele und Meilensteine fest, aber stellen Sie sicher, dass diese angepasst werden, wenn sich während des Entwicklungsprozesses Änderungen ergeben.

6. Automatisierung verwenden: Vermeide das Wiederholen manueller Routineaufgaben, indem du die Automatisierung und Skripting-Technologien für dein Projekt nutzt.
7. Tupelarbeit: Ermögliche den Entwicklern pairs oder Unit Arbeit auf einzelnen Funktionen. Dies minimiert die Zeit, die für die Entwicklung verloren geht.
8. Feedback einholen: Erheblich haperen Tests und Entwicklungszyklen, um Kundenfragen direkt zu adressieren.
9. Verwenden Sie Devops: DevOps-Techniken helfen, schneller und effizienter zu entwickeln, Integration und Testing-Ergebnisse zu reduzieren.
10. Kontinuierliche Auslieferung: Verliere keine Zeit und Empfänge den Lernerfolg, indem du die Code-Veröffentlichung regelmäßig in kleineren Einheiten über mehrere Benutzer oder Geräte auslieferst.

## DIE WICHTIGSTEN ERKENNTNISSE

### Grundlagen



Der Prozess der Erstellung eines Anforderungskatalogs in der Softwareentwicklung ist ein dynamischer Vorgang, der im Laufe der Entwicklung ständig ändert. Die Entwickler stehen hierbei vor großen Herausforderungen, denn um ein angemessenes Resultat zu erzielen, müssen die notwendigen Anforderungen klar und exakt definiert werden. Dieses White Paper wird darüber informieren, wie man einen Anforderungskatalog in der Softwareentwicklung aufstellt und wie man das dynamische Potenzial richtig nutzt.

### Anforderungen



Ein Anforderungskatalog ist eine systematische Liste aller notwendigen Funktionen, Komponenten, Anforderungen und Produkte, die für die Softwareentwicklung erforderlich sind. Es ist ein entscheidender Schlüssel, um das gewünschte Produkt zu entwickeln und es den Anforderungen des Kunden gerecht zu machen. Um einen Anforderungskatalog zu erstellen, müssen zuerst die zu lösenden Probleme definiert werden sowie das erreichbare Ergebnis. Anforderungen können sich auf technische, grafische, Usability, Performance- und Sicherheitsaspekte beziehen.

### Vorteile dynamischer Prozesse



Durch den dynamischen Ansatz können Entwickler eine Anforderung so anpassen, dass sie den Anforderungen des Kunden entspricht. Es ermöglicht es den Entwicklern, auf Kundenanforderungen zu reagieren und zu treffen, so dass bestehende Anforderungen angepasst oder neue Anforderungen hinzugefügt werden können. Da der Anforderungskatalog ständig angepasst werden kann, kann der Entwicklungsprozess effizienter gestaltet werden.



## Best Practices

Der Prozess der Erstellung eines Anforderungskatalogs für die Softwareentwicklung kann durch mehrere Prozesse und Strategien optimiert werden.

1. Involvierung der Kunden. Unabhängig davon, ob der Kunde über technische Erfahrung verfügt oder nicht, ist es wichtig, dass der Kunde bei der Erstellung des Anforderungskatalogs involviert wird, um die Anforderungen gerecht zu bewerten.
2. Organisieren und Priorisieren. Um einen übersichtlichen Anforderungskatalog zu erstellen, müssen die Anforderungen in Kategorien eingeteilt und nach Priorität geordnet werden. Dies erleichtert es den Entwicklern, den Prozess der Softwareentwicklung besser zu verstehen.
3. Verifizierung. Bevor eine Anforderung hinzugefügt oder bestehende Anforderungen geändert werden, muss verifiziert werden, ob die Anforderungen gültig sind und wie sie sinnvoll implementiert werden können.
4. Testen. Es ist wichtig, dass alle Anforderungen vor der Implementierung getestet werden, um sicherzustellen, dass sie alle Anforderungen des Kunden erfüllen.



## FAZIT

Der Erstellungsprozess eines Anforderungskatalogs in der Softwareentwicklung ist ein dynamischer Prozess, der viele Herausforderungen mit sich bringt. Ohne ein sorgfältig konzipiertes und strukturiertes Anforderungskatalog kann ein Entwickler nicht wissen, wie er das gewünschte Produkt erstellen soll. Um ein angemessenes Ergebnis zu erzielen, müssen die Anforderungen klar definiert und getestet werden. Der Zweck dieses White Papers war es, zu vermitteln, wie man einen Anforderungskatalog in der Softwareentwicklung erstellt und das dynamische Potenzial darin nutzt.





## Wichtige Schlussfolgerungen

1. **Flexibilität:** Schnelle Sprints ermöglichen es Teams, schnell auf Veränderungen zu reagieren und die benötigten Änderungen schnell vorzunehmen.
2. **Konzentration:** Schnelle Sprints helfen Teams, sich auf wichtige Aufgaben und Problemlösungen zu konzentrieren, statt sich auf unnötige Details zu konzentrieren.
3. **Hemmungslosigkeit:** Schnelle Sprints geben Teams eine Gelegenheit, Ideen und Techniken auszuprobieren, die sie vielleicht anderswo nicht versuchen würden.
4. **Kreativität:** Kürzere Sprints erleichtern es Teams, sich auf neue Ideen und Ansätze zu konzentrieren, mehr als vorher möglich.
5. **Fokus:** Mit kurzen Sprints ist es einfacher, sich auf bestimmte Ziele zu konzentrieren und zu überprüfen, ob sie erreicht werden.
6. **Schnellere Wiederaufnahme:** Teams müssen sich bei kurzen Sprints nicht mehr anhören als nötig. Dies ermöglicht es ihnen, schneller in die nächste Woche einzusteigen als es mit langen Sprints der Fall wäre.
7. **Längerer Produktzyklus:** Kurze Sprints ermöglichen es Teams, öfter als bisher Meilensteine und Produkte einzuführen und schneller mehr Nutzen für Kunden zu schaffen.