**Was ist das Spiralmodell?**

Laut WIKIPEDIA:

Das **Spiralmodell** ist ein [Vorgehensmodell](https://de.wikipedia.org/wiki/Vorgehensmodell_(Software)) in der [Softwareentwicklung](https://de.wikipedia.org/wiki/Softwareentwicklung), das im Jahr [1986](https://de.wikipedia.org/wiki/1986) von [Barry W. Boehm](https://de.wikipedia.org/wiki/Barry_W._Boehm) beschrieben wurde. Es ist ein [generisches](https://de.wikipedia.org/wiki/Generik)(d.h. es kann für alles benutzt werden) Vorgehensmodell und daher offen für bereits existierende Vorgehensmodelle. Das [Management](https://de.wikipedia.org/wiki/Manager_(Wirtschaft)) kann immer wieder eingreifen, da man sich spiralförmig voran entwickelt.

Das Spiralmodell fasst den Entwicklungsprozess im [Software-Engineering](https://de.wikipedia.org/wiki/Software-Engineering) in folgenden Phasen zusammen:

1. Festlegung von [Zielen](https://de.wikipedia.org/wiki/Ziel), Identifikation von [Alternativen](https://de.wikipedia.org/wiki/Alternative) und Beschreibung von Rahmenbedingungen
2. Evaluierung der [Alternativen](https://de.wikipedia.org/wiki/Alternative) und das Erkennen, Abschätzen und Reduzieren von [Risiken](https://de.wikipedia.org/wiki/Risiko), z. B. durch Analysen, Simulationen oder [Prototyping](https://de.wikipedia.org/wiki/Prototyping_(Softwareentwicklung)" \o "Prototyping (Softwareentwicklung))
3. Realisierung und Überprüfung des Zwischenprodukts
4. Planung des nächsten Zyklus der Projektfortsetzung.

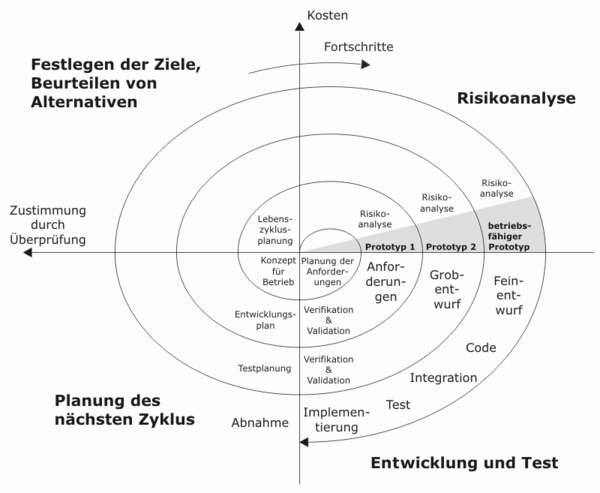
**Wozu braucht man das Spiralmodell?**

Das Spiralmodell beschreibt eine Möglichkeit wie man Projekte angehen kann.  
Wenn man zum Beispiel ein sehr komplexes Softwaresystem entwickelt, weiss man häufig zu Beginn noch nicht wie das endgültige System aussehen soll.  
Man möchte mit einem geringen Aufwand zuerst einmal ausprobieren, ob die Projektidee überhaupt umsetzbar ist.  
Mit den Ergebnissen des ersten Zyklus möchte man dann weiter arbeiten und weitere Funktionalität hinzufügen.

**Beispiel Inkrementelles Prototyping**

Beim Prototyping möchte man zuerst einmal schnell einen Prototyp erstellen.  
Prototypen sind nicht nur in der IT bekannt. Häufig werden z.B. Werkzeugmaschinen oder Autos als Prototypen erstellt.  
In der IT werden z.B. neue Webbrowser, Spiele, komplexe Backoffice Lösungen zuerst einmal als Prototyp erstellt.  
Die Systeme haben noch nicht alle Funktionalitäten implementiert.  
Der Kunde kann sich jedoch schon einmal ein Bild vom Endergebnis machen und kann entscheiden ob er mehr Geld in die Entwicklung stecken will.  
Der Betrieb kann sich ein Bild davon machen, ob das System später überhaupt betrieben werden kann.  
Der Entwickler kann “schnell” mal probieren, ob ein Lösungsweg zu einem befriedigenden Ergebnis führt oder ob man sich in einer Sackgasse befindet.  
Wenn der erste Prototyp steht, kann anhand der Erfahrungen auf diesen Prototyp aufgesetzt werden.  
Man durchläuft mehrmals die Schritte

* Festlegung der Ziele
* Beurteilung der Alternativen und Risiken
* Entwicklung und Test
* Planung des nächsten Zyklus

Unter http://www.mprove.de/script/05/im/\_media/Abb1Spiralmodell.png kann man folgendes Bild zum Spiralmodell finden:  


**Wie funktioniert dieses Modell?**

Wie schon erwähnt, beginnt man häufig mit einem ersten Prototypen.  
Ich erstelle zum Beispiel für meine Webapplikationskunden gerne erst einmal eine grobe Oberfläche und ein bisschen Basisfunktionalität die den Geschäftsprozess beschreibt.  
Mit dieser Oberfläche gehe ich zu meinen Kunden und erkläre ihnen wie sie mit dem fertigen Softwareprodukt später arbeiten können.  
Für die Erstellung des Prototyps benötige ich nur wenige Stunden und manchmal ein paar Tage.  
Der Kunde kann schon mal auf die einzelnen Elemente draufklicken und kann verstehen wie sich das Gesamtsystem später verhalten wird.  
Wie in der Grafik zu sehen, plane ich zuerst einmal den Lebenszyklus. Ich lege mit meinen Kunden fest, dass wir z.B. zwei Prototypen erstellen werden.  
Im ersten Prototyp wird zuerst einmal das Look and Feel dargestellt.  
Im Zweiten wird dann die tatsächliche Logik eingebaut.  
Wenn man sich im ersten Prototyp Zyklus befindet kann man schnell mal das grobe Gerüst “zusammenschustern” und sich dann an eine Risikobetrachtung machen. Man sollte sich fragen:

* Wird das Resultat dem Kunden überhaupt den gewünschten Nutzen bringen?
* Ist es sicher, dass man die Funktionalität auch erstellen kann?
* Sind die Kosten für den Aufbau wirklich gerechtfertigt?
* Gibt es andere Alternativen?
* Verbaut man sich mit diesem Prototyp den Weg für eine saubere Lösung?

Wenn man sich entschliesst, den Prototyp mit dem geplanten Funktionsumfang zu erstellen, dann erstellt man die notwendigen Komponenten. Man analysiert die notwendigen Anforderungen und setzt um.  
Der erste Zyklus ist hiermit abgeschlossen.  
Es folgt die Planung des nächsten Zyklus.

**Erstellt man immer gleich einen Prototypen?**

Nein.  
Bei komplexeren Systemen betrachtet man häufig zuerst die Risiken.  
Danach nimmt man die Anforderungen auf. Man designed die Lösung und schreibt alle notwendigen Betriebshandbücher. Wenn man sich vorstellen kann, dieses System später auch betreiben zu können, geht man in den zweiten Zyklus und erstellt den tatsächlichen Prototypen.

**Kann man denn ein Betriebskonzept schon zu Beginn schreiben?**

Das Betriebskonzept wird erst nach der Risikoanalyse und der Anforderungsanalyse erstellt.  
Beispielsweise kann bei der Anforderungsanalyse herauskommen, dass man eine Clusterumgebung benötigt. Dieser Cluster muss entsprechend mit RAID5 Storage aufgesetzt werden. Man benötigt weiterhin ein vernünftiges Backup- und Recovery Konzept.  
Mit diesen Angaben kann man durchaus ein Betriebshandbuch für den Cluster erstellen. Natürlich sollte man noch “schnell” ein entsprechendes Deploymentdiagramm und eine Serverauslegung durchführen.  
Dann kann man das System jedoch relativ schnell in Betrieb setzen und die Betriebshandbücher entsprechend mit den Kollegen vom Betrieb abklären.  
Auf den Clusterverbund kann man im zweiten Durchlauf dann die eigentliche Applikation aufsetzen.  
Mit dem Spiralmodell kann man sehr schön eine Sequenz von Aktionen beschreiben.

**Nachteile des Spiralmodells**

Wenn man das Spiralmodell, wie in der Grafik, zur Prototyperstellung verwendet, hat dieses Modell nicht nur Vorteile.  
Man setzt immer auf den schon bestehenden Prototyp auf. Fehler und schlechtes Design wird wiederverwendet und es entsteht normalerweise ein schlechtes Design, dass man nicht mehr korrigieren kann.  
Ein Prototyp sollte normalerweise nicht das Endprodukt sein. Leider sind viele Produkte aus dem Prototypstatus direkt in den Betriebsstatus überführt worden.  
Gründe hierfür sind die Kosten für eine korrekte Entwicklung, das System gibts ja schon und funktioniert usw.  
Mit dieser Vorgehensweise kann man leicht ein “Gebastel” in den produktiven Betrieb überführen.  
Auf der anderen Seite hat es auf alle Fälle seinen Scharm, wenn man relativ schnell etwas ausprobieren kann und dem Kunden schnell eine Lösung zaubern kann.  
In anderen Modellen, wie Beispielsweise dem Unified Process, findet man Iterationen vor, die mit dem Spiralmodell vergleichbar sind. Hier kann man also zu einem sehr frühen Zeitpunkt schon einen Prototyp erstellen und erst später die richtige Implementation liefern.  
Der Prototyp wird hier jedoch normalerweise nicht weiterentwickelt. Sobald man weiss wie und was man haben will, sollte man den Prototyp schnell vernichten.

Quellen:

<http://www.baldenhofer.eu/blog/it/kleine-erklaerung-zum-spiralmodell/>

<https://de.wikipedia.org/wiki/Spiralmodell>