**Spiralmodell**

**Wozu braucht man das Spiralmodell?**

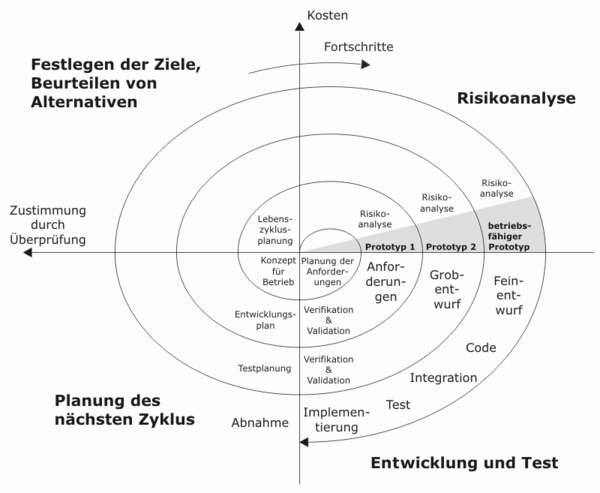
* Beschreibt Möglichkeit, wie man Projekte angehen kann
* Man kann mit bei komplexen Projekten mit geringen Aufwand herausfinden, on Idee überhaupt umsetzbar ist
* Mit Ergebnis des ersten Zyklus kann man gegebenenfalls weiterarbeiten

**Beispiel Inkrementelles Prototyping**

* Es wird ein Prototyp entwickelt
  + Nicht nur in der IT
  + In IT: Browser, Spiele
* Diese Systeme haben noch nicht alle Funktionen implementiert
* Kunde kann sich Bild vom System machen und danach entscheiden
* Betrieb kann sich Bild vom System machen und danach entscheiden
* Auf Prototyp kann aufgebaut werden

Man durchläuft mehrmals die Schritte

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



**Wie funktioniert dieses Modell?**

Es folgt die Planung des nächsten Zyklus.

* Man beginnt mit ersten Prototypen
* Man erstellt z.B bei Webapplikation Oberfläche mit Basisfunktionen
* Dieser Prototyp wird Kunden gezeigt, welcher eine Idee von fertigen Projekt bekommt. Er versteht wie sich das Gesamtsystem später verhalten wird

Grafik:

* Man macht mit Kunde aus, 2 Prototyps zu erstellen
* Bei ersten: wird das Look und Feel dargestellt
* Im zweiten: 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?

Wann ist Zyklus abgeschlossen?

* Wenn man sich entschliest den Prototyp mit geplanten Funktionen zu erstellen, mit notwendigen Komponenten. Durch analysieren.

**Erstellt man immer gleich einen Prototypen?**

Bei komplexen Systemen wird vorher sehr viel analysiertl.

**Nachteile des Spiralmodells**

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.  
Die Teile sind sonst ziemlich bissig und bleiben länger im Betrieb als man selbst

Ich hoffe, ich konnte ein wenig die Klarheiten beseitigen. Falls noch Fragen sind, könnt ihr euch ja mittels Kommentar melden.

* Man setzt auf bestehende Prototypen auf, Fehler und schlechtes design werden somit weitergegeben
* Der Prototyp sollte nicht direkt das Endprodukt sein, ist oft der Fall
  + Wegen Kosten und Leistungsminimierung