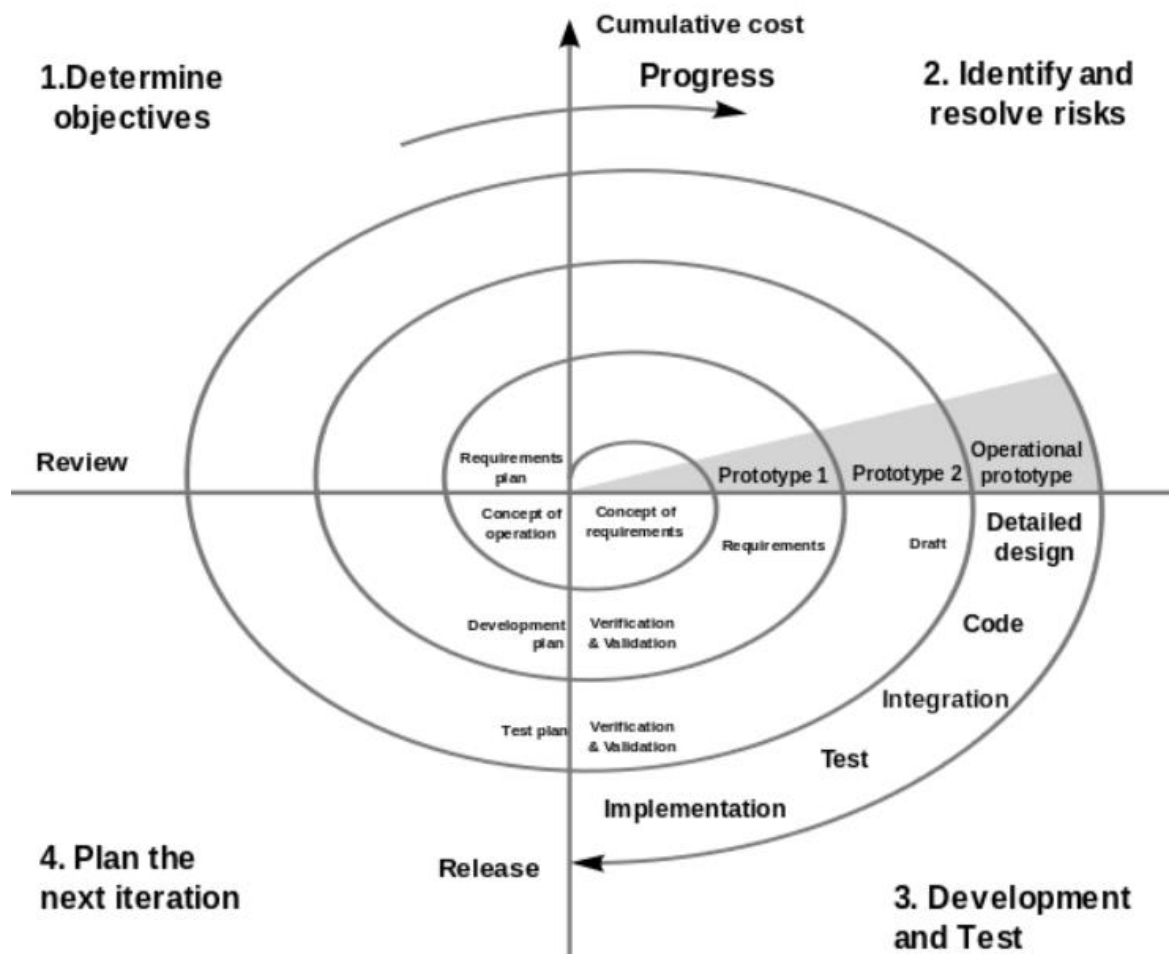


14. SPIRALNI MODEL

Barry Boehm je 1988. godine objavio razvoj sofvera kao "spiralni model", koji **kombinuje neke ključne aspekte modela vodopada, metodologije prototipa i RAD razvoj**, u nastojanju da iskombinuje prednosti ovih koncepata. Spiralni model stavlja naglasak u ključnoj oblasti koje su zapostavljene od strane drugih metodologija: iterativnom analizom rizika, posebno pogodnom za veliki broj kompleksnih sistema.

Osnovni principi spiralnog modela su:

- Fokus je na proceni rizika i smanjenju rizika projekta razbijanjem projekta na manje segmente i pružanje dosta jednostavnih promena tokom procesa razvoja, kao i pružanje prilike da ocene rizika vagaju razmatranje projekta tokom životnog ciklusa.
- Svaki ciklus uključuje progres istom sekvencom koraka, za svaki deo proizvoda i za svaki njegov nivo izrade, iz koncepta zahteva korisnika do kodiranja svakog individualnog programa.
- Svaki put oko spirale prelazi četiri osnovna kvadranta u kojima se izvršavaju osnovne aktivnosti:
 - (1) određivanje ciljeva, alternativa i ograničenja u iteraciji;
 - (2) analiza rizika - procena rizika, alternativa, identifikacija i rešavanje rizika;
 - (3) inženjering - razvoj novih nivoa softverskog proizvoda i verifikacija isporuke iz iteracije;
 - (4) planiranje sledeće iteracije uz procenu rezultata inženjeringa.
- Svaki ciklus treba početi sa identifikacijom zahteva i resursa, a na kraju svakog ciklusa završiti sa pregledima.



Funkcionisanje razvoja - Spiralni model razvoja softvera, sa svakom novom iteracijom, podrazumeva progresivni razvoj sve potpunije i potpunije verzije softvera. Tokom prvog ciklusa definišu se ciljevi, konceptualni zahtevi, (Concept of requirements), alternative i ograničenja, identifikuje i analizira rizik implementacije (Identify and resolve risks). Ukoliko analiza rizika inicira neizvesnost u zahtevima, tada se mogu raditi simulacije i procene modelima. Inženjering, tj. implementacija (Development and tests) zahteva se obavlja u svakom ciklusu spirale i to pristupom razvoja po modelu evolucijskog pristupa ili prototipskog razvoja (Prototype 1, Prototype 2, ... operational prototype). U svakoj sledećoj iteraciji broj aktivnosti raste kako se ciklusi udaljuju od centra spirale. Korisnik ocenjuje razvoj, tj. prototip i daje sugestije za modifikacije (Verification and validation). Na osnovu ulaznih podataka od korisnika, radi se sledeća faza procene i analize rizika, koja mora biti prisutna u svakom ciklusu. Na osnovu ove analize se donose odluke da li treba ili ne nastaviti sa daljim razvojem. Ukoliko je rizik prevelik, rad na razvoju se završava. Svaki novozapočeti ciklus spirale donosi kompletniji proizvod, ali i značajno više troškove razvoja. Svaki ciklus se završava ocenom (Review). Prvobitni koraci se izvršavaju sa visokim stepenom apstrakcije, a svaka naredna iteracija u spirali podrazumeva korake na nižem stepenu apstrakcije.

Prednosti modela se ogledaju u njegovoj fleksibilnosti za upravljanje softverskim inženjeringom, procena rizika se vrši na svakom nivou apstrakcije, a model se prilagođava svakoj kombinaciji različitih pristupa u razvoju softvera.

Nedostaci modela se ogledaju u nepostojanje veze sa standardima, problem predstavlja neotkrivanje rizika.

Ovo je, trenutno, **najrealniji model razvoja softvera za velike sisteme**, ali nije dovoljno primenjivan i prihvaćen.