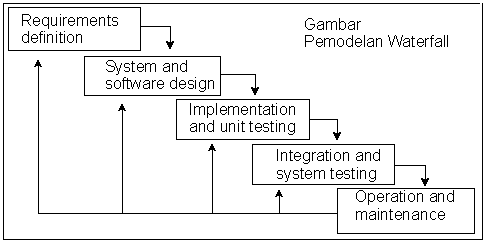
1. Waterfall Model

Waterfall Model adalah sebuah metode pengembangan software yang bersifat sekuensial. Metode ini dikenalkan oleh Royce pada tahun 1970 dan pada saat itu disebut sebagai siklus klasik dan sekarang ini lebih dikenal dengan sekuensial linier. Selain itu Model ini merupakan model yang paling banyak dipakai oleh para pengembang software. Inti dari metodewaterfall adalah *pengerjaan dari suatu system dilakukan secara berurutan atau secara linear*. Jadi jika langkah satu belum dikerjakan maka tidak akan bisa melanjutkan kelangkah 2, 3 dan seterusnya.  Secara otomatis tahapan ke-3 akan bisa dilakukan jika tahap ke-1 dan ke-2 sudah dilakukan.



1. Analisa kebutuhan (Requirement Analysis)/(Requirements analysis and definition)

Langkah ini merupakan analisa terhadap kebutuhan sistem. Pengumpulan data dalam tahap ini bisa dilakukan dengan sebuah penelitian, wawancara atau study literatur. Seorang system analis akan menggali informasi sebanyak-banyaknya dari user sehingga akan tercipta sebuah system komputer yang bisa melakukan tugas-tugas yang diinginkan oleh user tersebut. Tahapan ini akan menghasilkan dokumen user requirement atau bisa dikatakan sebagai data yang berhubungan dengan keinginan user dalam pembuatan sistem. Dokumen inilah yang akan menjadi acuan system analis untuk menterjemahkan kedalam bahasa pemprogram.

1. Design sistem (System Design)

Proses desain akan menerjemahkan syarat kebutuhan ke sebuah perancangan perangkat lunak yang dapat diperkirakan sebelum dibuat coding. Proses ini berfokus pada struktur data, arsitektur perangkat lunak, representasi interface, dan detail (algoritma) prosedural. Tahapan ini akan menghasilkan dokumen yang disebut software requirment. Dokumen inilah yang akan digunakan programmer untuk melakukan aktivitas pembuatan sistemnya.

1. Coding & Testing/penulisankode Program (Implementation)

Coding merupakan penerjemahan design dalam bahasa yang bisa dikenali oleh komputer.Dilakukan oleh programmer yang akan meterjemahkan transaksi yang diminta oleh user. Tahapan inilah yang merupakan tahapan secara nyata dalam mengerjakan suatu sistem.Dalam artian penggunaan computer akan dimaksimalkan dalam tahapan ini. Setelah pengkodean selesai maka akan dilakukan testing terhadap sistem yang telah dibuat tadi. Tujuan testingadalah menemukan kesalahan-kesalahan terhadap system tersebut dan kemudian bisa diperbaiki.

1. Penerapan / pengujian program (Integration & Testing)

Tahapan ini bisa dikatakan final dalam pembuatan sebuah sistem.Setelah melakukan analisa, design dan pengkodean maka sistem yang sudah jadiakan digunakan oleh user.

1. Pemeliharaan (Operation & Maintenance)

Perangkat lunak yang sudah disampaikan kepada pelanggan pasti akan mengalami perubahan. Perubahan tersebut bisa karena mengalami kesalahan karena perangkat lunak harus menyesuaikan dengan lingkungan (periperal atau system operasi baru) baru, atau karena pelanggan membutuhkan perkembangan fungsional.

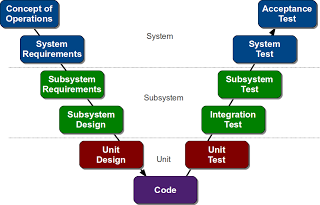
Kelebihan dan kelemahan waterfall model:

* Kelebihan Waterfall Model:
  + Kualitas dari sistem yang dihasilkan akan baik. Ini dikarenakan oleh pelaksanaannya secara bertahap. Sehingga tidak terfokus pada tahapan tertentu.
  + Document pengembangan system sangat terorganisir, karena setiap fase harus terselesaikan dengan lengkap sebelum melangkah ke fase berikutnya. Jadi  setiap fase atau tahapan akan mempunyai dokumen tertentu.
  + Metode ini masih lebih baik digunakan walaupun sudah tergolong kuno, daripada menggunakan pendekatan asal-asalan. Selain itu, metode ini juga masih masuk akal jika kebutuhan sudah diketahui dengan baik.
* Kelemahan Waterfall Model:
  + Diperlukan majemen yang baik, karena proses pengembangan tidak dapat dilakukan secara berulang sebelum terjadinya suatu produk.
  + Kesalahan kecil akan menjadi masalah besar jika tidak diketahui sejak awal pengembangan yang berakibat pada tahapan selanjutnya.
  + Pelanggan sulit menyatakan kebutuhan secara eksplisit sehingga tidak dapat mengakomodasi ketidak pastian pada saat awal pengembangan.
  + Pelanggan harus sabar, karena pembuatan perangkat lunak akan dimulai ketika tahap desain sudah selesai. Sedangkan pada tahap sebelum desain bisa memakan waktu yang lama.
  + Pada kenyataannya, jarang mengikuti urutan sekuensial seperti pada teori. Iterasi sering terjadi menyebabkan masalah baru.

1. V-shapes model

Model ini merupakan perluasan dari model waterfall. Disebut sebagai perluasan karena tahap-tahapnya mirip dengan yang yang dalam model waterfall. Jika dalam model waterfall proses dijalankan secara linier, maka dalam model V proses dalikukan bercabang dalam model V ini digambarkan hubungan antara tahap pengembangan software dengan tahap pengujiannya.

 Bisa dikatakan model ini merupakan perluasan dari model waterfall. Disebut sebagai perluasan karena tahap-tahapnya mirip dengan yang terdapat dalam model waterfall. Jika dalam model waterfall proses dijalankan secara linear, maka dalam model V proses dilakukan bercabang. Dalam model V ini digambarkan hubungan antara tahap pengembangan software dengan tahap pengujiannya.



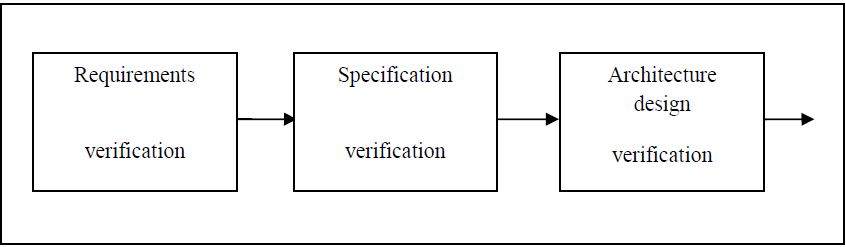
Kelebihan dan kekurangan V-shapes model:

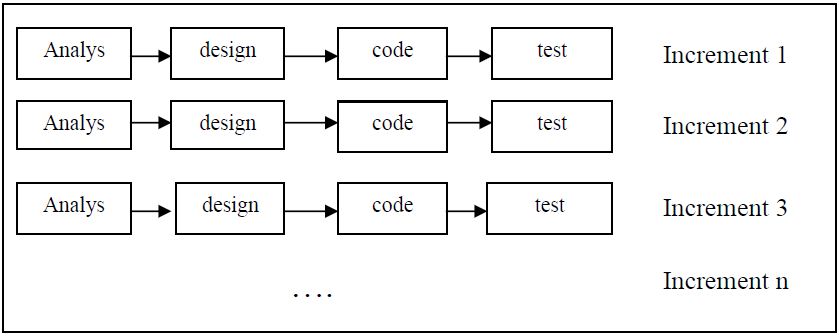
* Keuntungan V Model
  + Bahasa yang digunakan untuk merepresentasikan konsep V model menggunakan bahasa formal. Contoh : dengan menggunakan objek model ataupun frame-frame • Meminimalisasikan kesalahan pada hasil akhir karena ada test pada setiap prosesnya
  + Penyesuaian yang cepat pada projek yang baru
  + Memudahkan dalam pembuatan dokumen projek
  + Biaya yang murah dalam perawatan dan modifikasinya
  + V Model sangat fleksibel. V Model mendukung project tailoring dan penambahan dan pengurangan method dan tool secara dinamik. Akibatnya sangat mudah untuk melakukan tailoring pada V Model agar sesuai dengan suatu proyek tertentu dan sangat mudah untuk menambahkan method dan tool baru atau menghilangkan method dan tool yang dianggap sudah obsolete.
  + V Model dikembangkan dan di-maintain oleh publik. User dari V Model berpartisipasi dalam change control board yang memproses semua change request terhadap V Model.
* Kerugian V Model
  + Aktifitas V-Model hanya difokuskan pada projectnya saja, bukan pada keseluruhan organisasi. V-Model adalah proses model yang hanya dikerjakan sekali selama project saja, bukan keseluruhan organisasi.
  + Prosesnya hanya secara sementara. Ketika project selesai, jalannya proses model dihentikan. Tidak berlangsung untuk keseluruhan organisasi.
  + Metode yang ditawarkan terbatas. Sehingga kita tidak memiliki cara pandang dari metode yang lain. Kita tidak memiliki kesempatan untuk mempertimbangkan jika ada tools lain yang lebih baik.
  + Toolnya tidak selengkap yang dibicarakan. SDE (Software Development Environment).Tidak ada tools untuk hardware di V-Model. Tool yang dimaksud adalah “software yang mendukung pengembangan atau pemeliharaan / modifikasi dari system IT.
  + V Model adalah model yang project oriented sehingga hanya bisa digunakan sekali dalam suatu proyek.
  + V Model terlalu fleksibel dalam arti ada beberapa activity dalam V Model yang digambarkan terlalu abstrak sehingga tidak bisa diketahui dengan jelas apa yang termasuk dalam activity tersebut dan apa yang tidak.

1. Incremental Model

Incremental model adalah model pengembangan sistem pada software engineering berdasarkan requirement software yang dipecah menjadi beberapa fungsi atau bagian sehingga model pengembangannya secara bertahap. dilain pihak ada mengartikan model incremental sebagai  perbaikan dari model waterfall dan sebagai standar pendekatan topdown. Layaknya Model Waterfall, model ini pun juga memiliki tahapan tahapan untuk perancangan perangkat lunaknya, yaitu:

1. Requirement , Requirment adalah proses tahapan awal yang dilakukan pada incremental model adalah penentuan kebutuhan atau analisis kebutuhan.
2. Specification, Specification adalah proses spesifikasi dimana menggunakan analisis kebutuhan sebagai acuannya.
3. Architecture Design, adalah tahap selanjutnya, perancangan software yang terbuka agar dapat diterapkan sistem pembangunan per-bagian pada tahapan selanjutnya.
4. Code setelah melakukan proses desain selanjutnya ada pengkodean.
5. Test merupakan tahap pengujian dalam model ini.





Tahapan-tahapan tersebut dilakukan secara berurutan. Setiap bagian yang sudah selesai dilakukan testing, dikirim ke pemakai untuk langsung dapat digunakan. Pada incremental model, tiga tahapan awal harus diselesaikan terlebih dahulu sebelum sebelum tahap membangun tiap increment. Untuk mengantisipasi kondisi yang terjadi pada incremental model, diperkenalkan model More Risky Incremental Model. Model ini menerapkan sistem kerja yang paralel. Setelah daftar kebutuhan didapatkan dari pemakai, tim spesifikasi membuat spesifikasi untuk modul pertama. Setelah spesifikasi pertama selesai, tim desain menindak lanjuti. Tim spesifikasi sebelumnya juga langsung membuat spesifikasi untuk model kedua, dan seterusnya. Jadi, tidak harus menunggu modul pertama selesai hingga dikirim ke user.

Beberapa Kelebihan Dari Mode Incremental atara lain :

* Merupakan model dengan manajemen yang sederhana
* Pengguna tidak perlu menunggu sampai seluruh sistem dikirim untuk mengambil keuntungan dari sistem tersebut. Increment yang pertama sudah memenuhi persyaratan mereka yang paling kritis, sehingga perangkat lunak dapat segera digunakan.
* Resiko untuk kegagalan proyek secara keseluruhan lebih rendah. Walaupun masalah masih dapat ditemukan pada beberapa increment. Karena layanan dengan prioritas tertinggi diserahkan pertama dan increment berikutnya diintegrasikan dengannya, sangatlah penting bahwa layanan sistem yang paling penting mengalami pengujian yang ketat. Ini berarti bahwa pengguna akan memiliki kemungkinan kecil untuk memenuhi kegagalan perangkat lunak pada increment sistem yang paling bawah.
* Nilai penggunaan dapat ditentukan pada setiap increment sehingga fungsionalitas sistem disediakan lebih awal.
* Memiliki risiko lebih rendah terhadap keseluruhan pengembagan sistem,
* Prioritas tertinggi pada pelayanan sistem adalah yang paling diuji

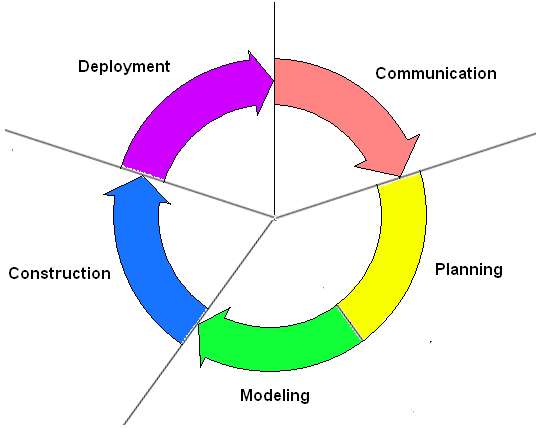
Kelemahannya adalah :

* kemungkinan tiap bagian tidak dapat diintegrasikan
* Dapat menjadi build and Fix Model, karena kemampuannya untuk selalu mendapat perubahan selama proses rekayasa berlangsung
* Harus Open Architecture
* Mungkin terjadi kesulitan untuk memetakan kebutuhan pengguna ke dalam rencana spesifikasi masing-masing hasil increment.

1. Prototyping Model

Prototyping adalah pengembangan yang cepat dan pengujian terhadap model kerja (prototipe) dari aplikasi baru melalui proses interaksi dan berulang-ulang yang biasa digunakan ahli sistem informasi dan ahli bisnis. Prototyping disebut juga desain aplikasi cepat (rapid application design/RAD) karena menyederhanakan dan mempercepat desain sistem (O'Brien, 2005).

Sebagian user kesulitan mengungkapkan keinginannya untuk mendapatkan aplikasi yang sesuai dengan kebutuhannya. Kesulitan ini yang perlu diselesaikan oleh analis dengan memahami kebutuhan user dan menerjemahkannya ke dalam bentuk model (prototipe). Model ini selanjutnya diperbaiki secara terus menerus sampai sesuai dengan kebutuhan user.



Berikut adalah tahapan dalam prototyping model:

1. Pengumpulan Kebutuhan. Pada tahap pengumpulan kebutuhan, costumer memaparkan apa saja yang dibutuhkan dalam sistem yang akan dibuat.
2. Membangun Prototyping. Pada tahap pembangunan prototyping, pengembang dan pembuat sistem bersama-sama membuat format input maupun  output yang akan dihasilkan oleh sistem yang dibuat.
3. Evaluasi protoptyping  
   Evaluasi ini dilakukan oleh pelanggan apakah prototyping yang sudah dibangun sudah sesuai dengan keinginan pelanggan. Jika sudah sesuai maka langkah keempat akan diambil. Jika tidak, maka prototyping diperbaiki dengan mengulang langkah 1, 2 , dan 3.
4. Mengkodekan system  
   Dalam tahap ini prototyping yang sudah disepakati diterjemahkan ke dalam bahasa pemrograman yang sesuai.
5. Menguji system  
   Setelah sistem sudah menjadi suatu perangkat lunak yang siap pakai, harus dites dahulu sebelum digunakan. Pengujian ini dilakukan dengan White Box, Black Box, Basis Path, pengujian arsitektur dan lain-lain.
6. Evaluasi Sistem  
   Pelanggan mengevaluasi apakah sistem yang sudah jadi sudah sesuai dengan yang diharapkan . Jika sudah, maka langkah ketujuh dilakukan, jika belum maka mengulangi langkah 4 dan 5.
7. Menggunakan system  
   Perangkat lunak yang telah diuji dan diterima pelanggan siap untuk digunakan

Keunggulan prototyping :

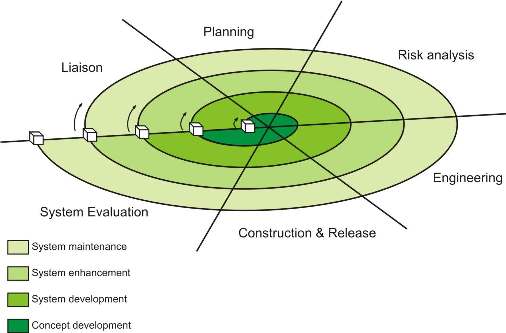
* Komunikasi akan terjalin baik antara pengembang dan pelanggan.
* Pengembang dapat bekerja lebih baik dalam menentukan kebutuhan setiap pelanggannya.
* Pelanggan berperan aktif dalam proses pengembangan sistem.
* Lebih menghemat waktu dalam pengembangan sistem.
* Penerapan menjadi lebih mudah karena pemakai mengetahui apa yang diharapkannya.

Kelemahan prototyping :

* Pelanggan kadang tidak melihat atau menyadari bahwa perangkat lunak yang ada belum mencantumkan kualitas perangkat lunak secara keseluruhan dan juga belum memikirkan kemampuan pemeliharaan untuk jangka waktu lama.
* Pengembang biasanya ingin cepat menyelesaikan proyek sehingga menggunakan algoritma dan bahasa pemrograman yang sederhana untuk membuat prototyping lebih cepat selesai tanpa memikirkan lebih lanjut bahwa program tersebut hanya merupakan sebuah kerangka kerja(blueprint) dari sistem .
* Hubungan pelanggan dengan komputer yang disediakan mungkin tidak mencerminkan teknik perancangan yang baik dan benar.

1. Spiral Model

Model spiral pada awalnya diusulkan oleh Boehm, adalah model proses perangkat lunak evolusioner yang merangkai sifat iteratif dari prototype dengan cara kontrol dan aspek sistematis model sequensial linier.  
        Model iteratif ditandai dengan tingkah laku yang memungkinkan pengembang mengembangkan versi perangkat lunak yang lebih lengkap secara bertahap. Perangkat lunak dikembangkan dalam deretan pertambahan. Selama awal iterasi, rilis inkremantal bisa berupa model/prototype kertas, kemudian sedikit demi sedikit dihasilkan versi sistem yang lebih lengkap.



Tahapan-Tahapan Model Spiral  
Model spiral dibagi menjadi enam wilayah tugas yaitu:

* 1. Komunikasi pelanggan    
         Yaitu tugas-tugas untuk membangun komunikasi antara pelanggan dan kebutuhankebutuhan yang diinginkan oleh pelanggan
  2. Perencanaan  
        Yaitu tugas-tugas untuk mendefinisikan sumber daya, ketepatan waktu, dan proyek informasi lain yg berhubungan.
  3. Analisis Resiko  
         Yaitu tugas-tugas yang dibutuhkan untuk menaksir resikomanajemen dan teknis.
  4. Perekayasaan  
         Yaitu tugas yang dibutuhkan untuk membangun satu atau lebih representasi dari  
         apikasi tersebut.
  5. Konstruksi dan peluncuran  
         Yaitu tugas-tugas yang dibutuhkan untuk mengkonstruksi, menguji, memasang , dan memberi pelayanan kepada pemakai.
  6. Evaluasi Pelanggan  
         Yaitu tugas-tugas untuk mendapatkan umpan balik dari pelanggan.

Kelebihan model Spiral :

1. Dapat disesuaikan agar perangkat lunak bisa dipakai selama hidup perangkat lunak komputer.
2. Lebih cocok untuk pengembangan sistem dan perangkat lunak skala besar.
3. Pengembang dan pemakai dapat lebih mudah memahami dan bereaksi terhadap resiko setiap tingkat evolusi karena perangkat lunak terus bekerja selama proses.
4. Menggunakan prototipe sebagai mekanisme pengurangan resiko dan pada setiap keadaan di dalam evolusi produk.
5. Tetap mengikuti langkah-langkah dalam siklus kehidupan klasik dan memasukkannya ke dalam kerangka kerja iteratif.
6. Membutuhkan pertimbangan langsung terhadp resiko teknis sehingga mengurangi resiko sebelum menjadi permaslahan yang serius.

Kelemahan model Spiral :

1. Sulit untuk menyakinkan pelanggan bahwa pendekatan evolusioner ini bisa dikontrol.
2. Memerlukan penaksiran resiko yang masuk akal dan akan menjadi masalah yang serius jika resiko mayor tidak ditemukan dan diatur.
3. Butuh waktu lama untuk menerapkan paradigma ini menuju kepastian yang absolut.