

Konsep Dasar Perangkat Lunak

Materi 1

Konsep Dasar Perangkat Lunak

Outlines:

- Arti dan Definisi Perangkat Lunak
- Jenis-jenis Perangkat Lunak
- Tanggungjawab Profesional dan Etika
- Siklus Hidup Perangkat Lunak

Materi 1

Konsep Dasar Perangkat Lunak

Sasaran:

- Siswa memahami arti dan pentingnya perangkat lunak.
- Siswa memahami tanggungjawab profesional dan etika didalam perekayasaan perangkat lunak.
- Siswa memahami tentang System Development Lifecycle dan jenis-jenisnya

Referensi:

- Pressman, Roger S., Software Engineering : A Practitioner's Approach, 6th edition, McGraw-Hill International, 2005
- Sommerville, Ian, Software Engineering, 7th Addison Wesley Publishing Company, 2003
- Whitten, Jeffry L. and Bentley, Lonnie D. Introduction to System analysis and Design. McGraw-Hill International, 2005

Apa itu Perangkat Lunak?

Program adalah sekumpulan instruksi yang berupa kode yang dimengerti oleh komputer untuk melakukan pekerjaan tertentu.

Software adalah sekumpulan program-program untuk tujuan tertentu yang didalamnya terdapat **dokumentasi untuk menginstall, menggunakan, mengembangkan dan melakukan perawatan** terhadap program tersebut. (Sommerville, 1992)

Software adalah sebuah produk yang dibangun secara profesional dan terdapat dukungan jangka panjang yang berisi informasi tentang arsitektur, cakupan, isi (program) dan **dokumentasi dalam bentuk kertas maupun data digital** (Pressman, 2005).

Apa itu Perangkat Lunak? (2)

Software engineering adalah sebuah disiplin ilmu dalam bidang teknik yang berfokus masalah-masalah praktis dalam pengembangan sistem perangkat lunak yang besar. (Sommerville, 1992)

Seorang software engineer tidak hanya pada aspek programming atau ilmu komputer saja, namun mereka harus **profesional** dalam menggunakan **teori-teori pendukung** dan dipraktekkan secara **cost-effectively** untuk menyelesaikan masalah.

Software sebuah produk

- a. Engineer/Analist = program, content (data), dan dokumentasi
- b. User/ Konsumen = sebuah produk/penyedia informasi yang membuat **kebutuhan user** lebih mudah.

Ingat!

Engineer/analist bekeja berdasar/berorientasi pada kebutuhan/keinginan user, tidak bekerja sesuai idealisme mereka belaka, namun seorang engineer/analist menawarkan solusi yang paling ideal bagi kebutuhan user.

Jenis-Jenis Perangkat Lunak

System software

program yang dibuat untuk melayani program lain (compiler, dll)

Application software

Aplikasi independen untuk menyelesaikan tugas yang spesifik

Engineering/scientific software

software untuk keperluan pengetahuan (orbit planet, vulkanologi, simulasi dll)

Embedded software

software untuk memonitor dan mengontrol perangkat yang terintegrasi dengan itu (sistem skala bahan bakar pada kendaraan, speedometer dll)

Jenis-Jenis Perangkat Lunak (2)

Product-line software

software dalam lingkungan tertentu untuk melakukan beberapa task (komponen office(word prosessor, spreadsheet, entertainment, grafis dll))

Web-applications

hypertext document yang dipublikasikan melaui internet..

Artificial intelligence software

software yang memiliki kecerdasan (prediksi, pengambilan keputusan, rekognisi dll, misal game, robot identifikasi voice maupun image)

Tantangan Perangkat Lunak

Beberapa perangkat lunak yang menjadi trend kedepan:

- Ubiquitous computing
- Opensource
- The new economy
- Komputasi awan

Tugas → cari tahu ttng dropbox.com, mendeley.com, salesforce.com, Google engine, Amazon EC2, comapping.com, sme.telkom.co.id

Tanggungjawab Profesional dan Etika

Berapa anda belanja software dalam setahun ini..?

Bagaimana anda mendapatkan software yang anda gunakan saat ini?

Apakah anda pernah melakukan *cracking* ?

Apa yang anda ketahui tentang hak cipta?

Tanggungjawab Profesional dan Etika (2)

Bagaimana tanggungjawab anda terhadap project yang telah anda buat/sepakati (eksternal)?

Layanan apa saja yang telah anda hadirkan dalam mengakomodasi kebutuhan konsumen?

Apa yang akan anda lakukan ketika terdapat komplain dari konsumen?

Bagaimana anda menjalin hubungan mutualisme dengan konsumen?

Tanggungjawab Profesional dan Etika (3)

ACM menefinisikan 8 kode etik bagi seorang software engineer:

1. Public

→ Konsisten terhadap perhatian umum (eksternal)

2. Client and employer

→ Konsisten terhadap perhatian client & pegawai (internal)

3. Product

→ Memastikan produk berkualitas, berstandar dan cost-effective

4. Judgment

→ Memiliki pertimbangan yang matang dan keputusan yg berkualitas

Tanggungjawab Profesional dan Etika (4)

ACM menefinisikan 8 kode etik bagi seorang software engineer:

5. Management

→ Menjadi pemimpin yang baik, memahami management, policies dan prosedur yang berlaku

6. Profession

→ Profesional dengan pekerjaannya

7. Colleagues

→ fair dan saling mendukung antar kolega

8. Self

→ selalu berbenah diri, menambah pengetahuan dan skill

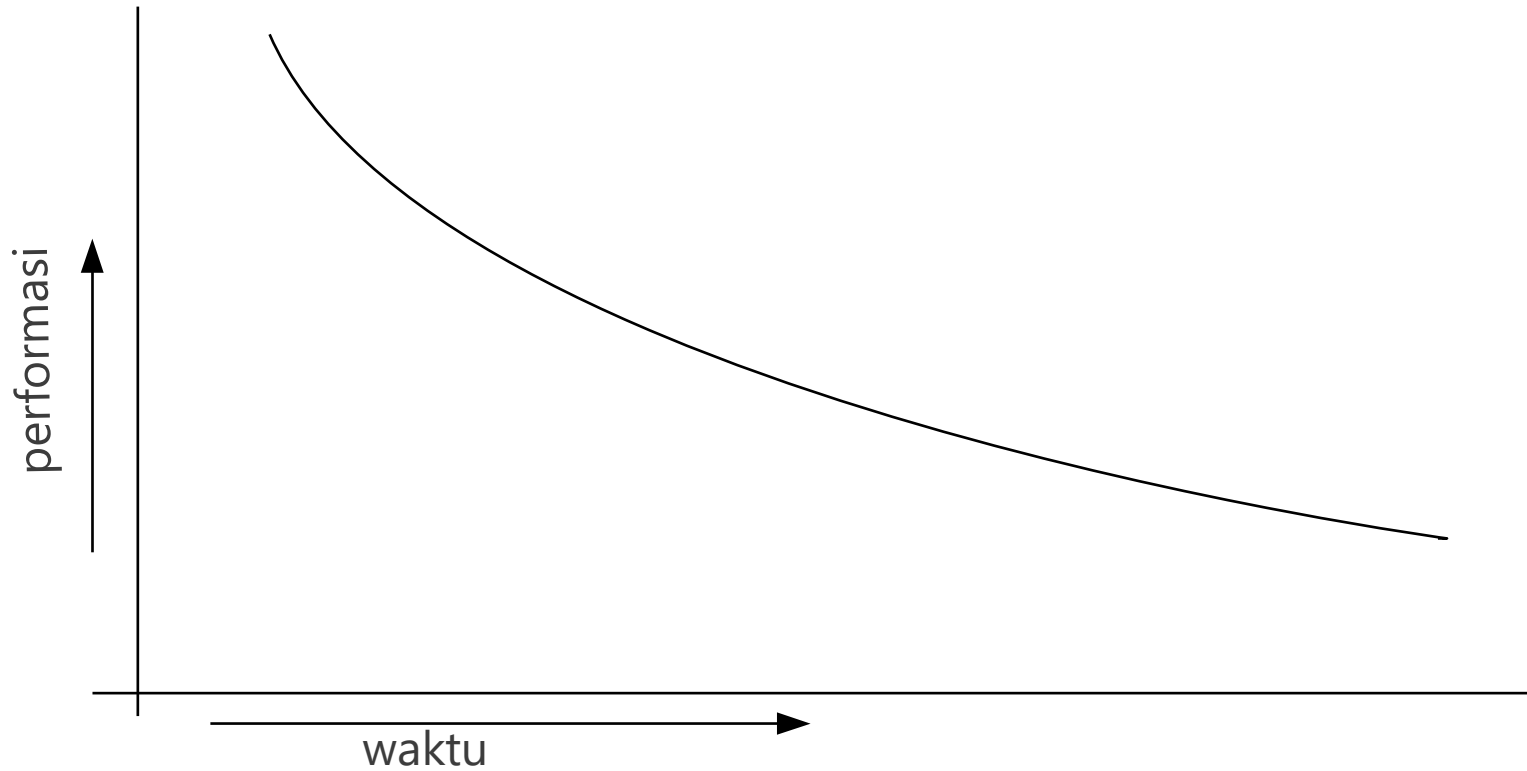
Sumber: <http://www.acm.org/about/se-code>

Siklus hidup perangkat lunak

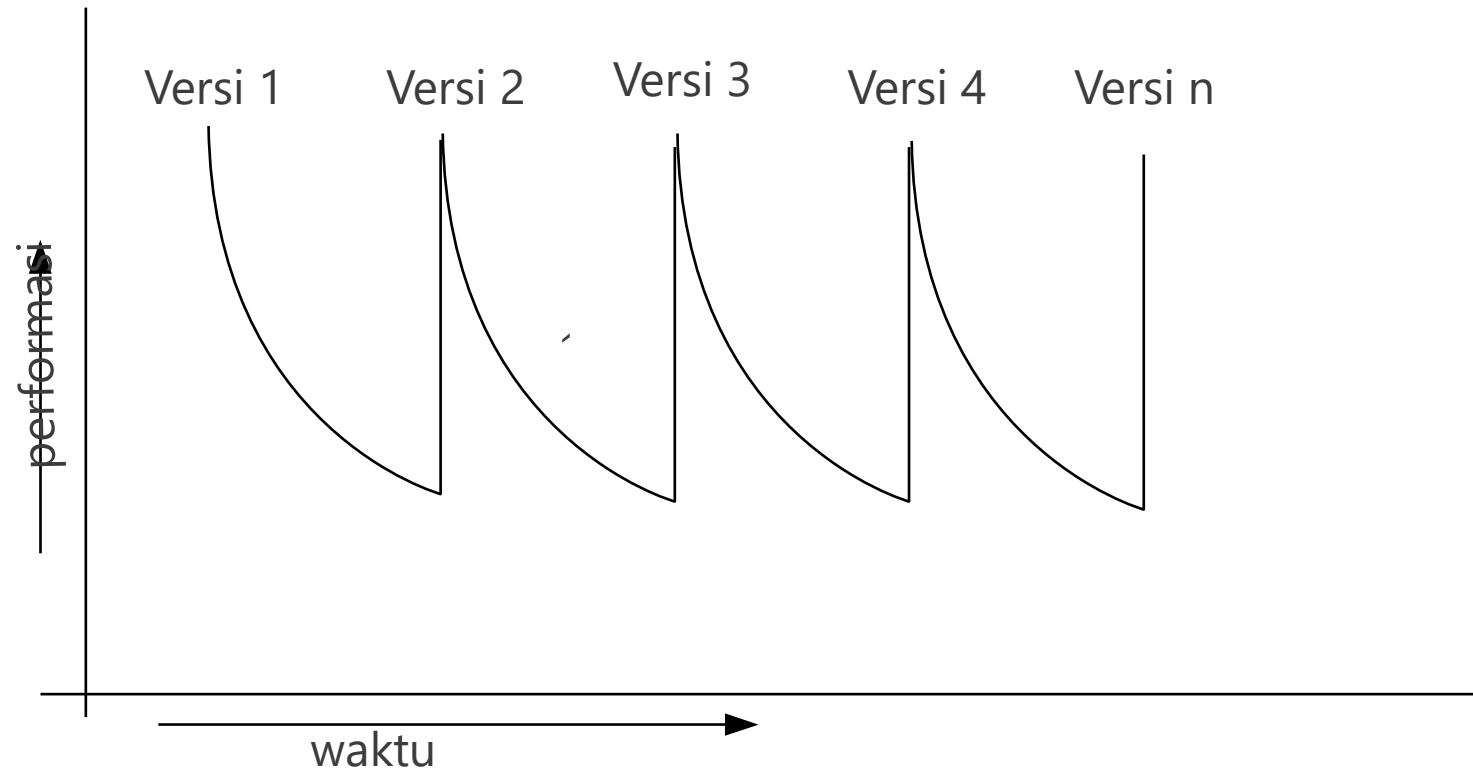
Siklus hidup perangkat lunak memiliki perbedaan yang mendasar dengan perangkat keras.

- Hardware/perangkat keras memiliki masa usang: kapasitas, performasi, ukuran, bentuk dan trend.**
- Software tidak memiliki masa usang. Software bisa dirombak (total maupun sebagian) yang disesuaikan dengan kebutuhan.**

Siklus hidup perangkat lunak (2)



Siklus hidup perangkat lunak (2)



Latihan..

1. Apa perbedaan antara program dengan software?
2. Bagaimana seharusnya seorang engineer dalam mengembangkan sistem?
3. Sebutkan 2 sudut pandang yang harus diperhatikan oleh seorang engineer/analyst!
4. Apa perbedaan antara opensource dengan komersial, beri contoh!
5. Sebutkan kode etik bagi seorang engineer!
6. Bagaimana siklus hidup perangkat lunak?

System Development Lifecycle (SDLC)

System Development Life Cycle (SDLC) adalah tahapan-tahapan produktif dalam pengembangan sistem (perangkat lunak)

Tahapan produktif bukan dari banyaknya pertemuan, namun kemajuan yang terukur. Biasanya diwujudkan dengan kemajuan besar dalam sebuah proses yang telah dijadwalkan sebelumnya.

Jenis-Jenis SDLC

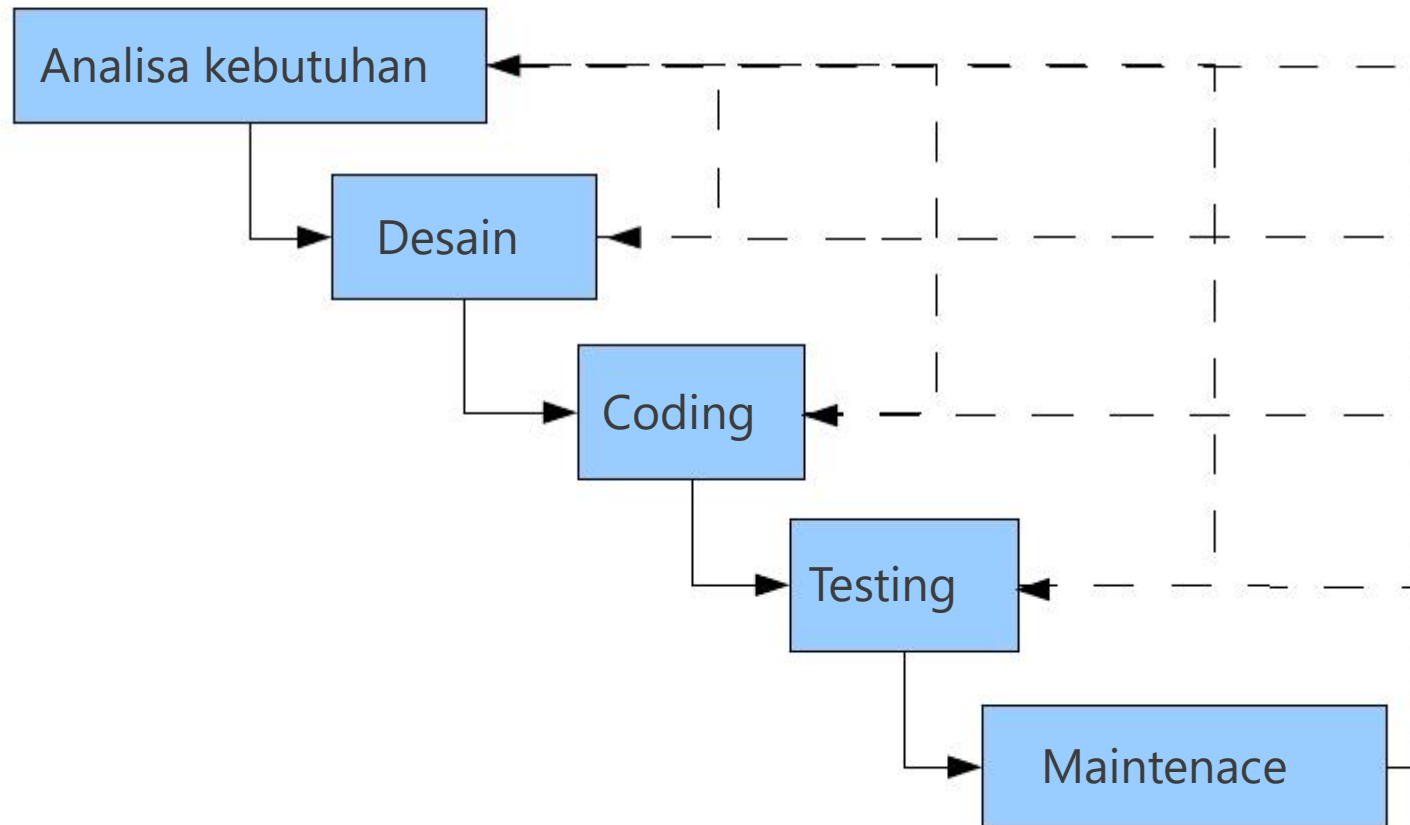
Model air terjun (waterfall)

- Salah satu model dari SDLC yang bersifat linear.
- Proses pengembangan bertahap dan teratur
- Output tahap sebelumnya menjadi input tahap berikutnya.(Kristanto, 2004)
- Dokumentasi proses sangat lengkap

Tahap-tahapnya

1. Analisa kebutuhan
2. Desain sistem dan software
3. Implementasi dan *coding*
4. Integrasi dan testing system
5. Operasional dan perawatan (opsional)

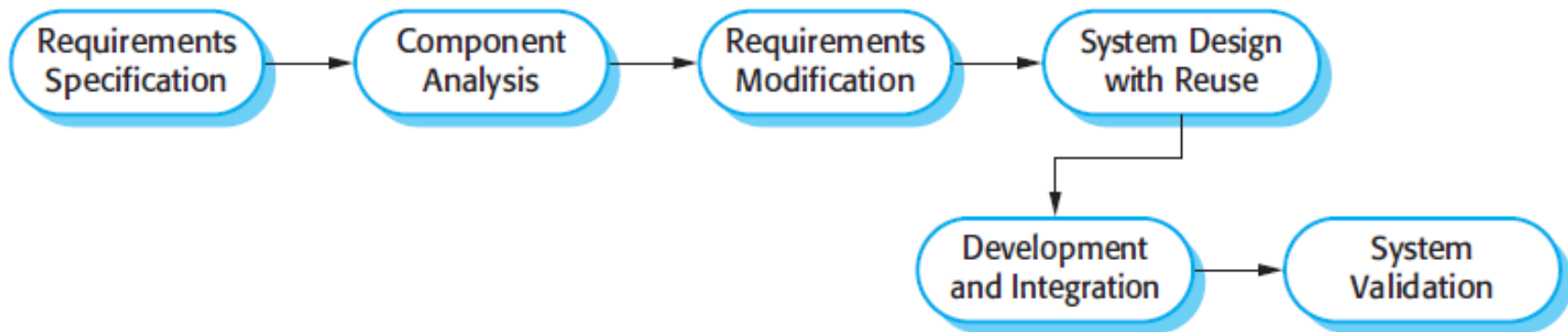
Jenis-Jenis SDLC - Waterfall



Jenis-Jenis SDLC (2)

Model Re Used Oriented

Pendekatan ini didasarkan pada keberadaan sejumlah besar komponen reusable. Proses pengembangan sistem berfokus pada mengintegrasikan komponen ini ke dalam sistem baru daripada mengembangkan sistem dari awal.

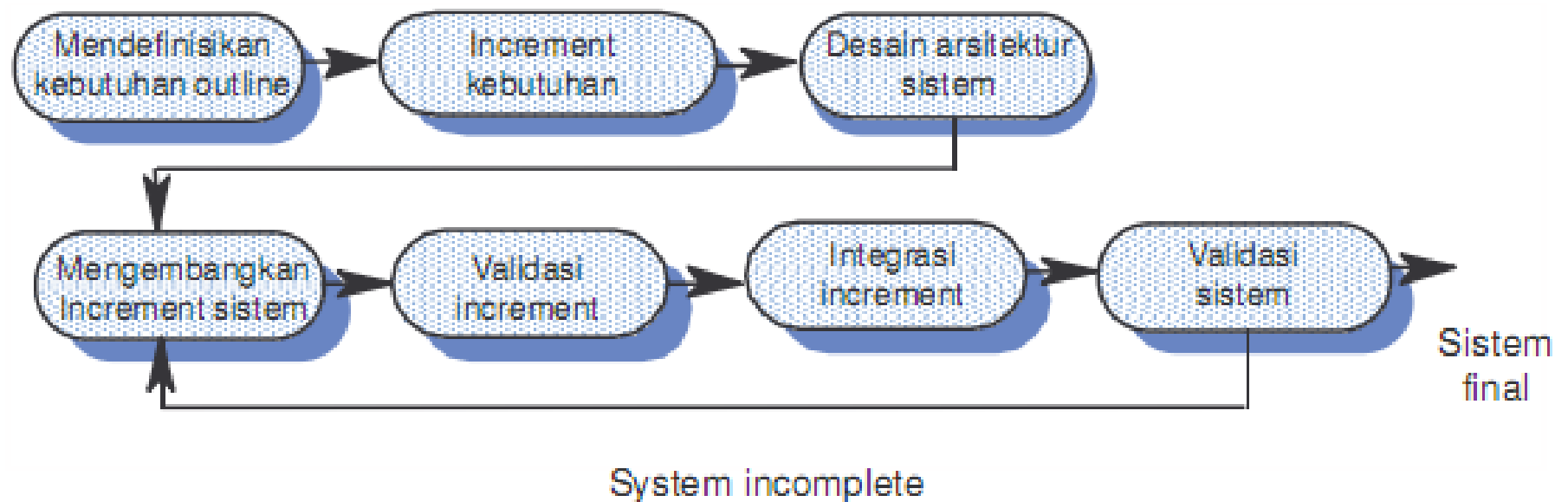


Jenis-Jenis SDLC (2)

Model Incremental Development

- Delivery sistem bukan pengiriman tunggal, tetapi pengembangan dan pengiriman dipecah menjadi bertahap dengan setiap 'increment' memberikan bagian fungsional yang diperlukan.
- Persyaratan user diprioritaskan dan kebutuhan prioritas tertinggi akan di masukan ke dalam 'increment' awal.
- Setelah pengembangan 'increment' dimulai, persyaratan dibekukan lebih dahulu dan persyaratan untuk 'increment' selanjutnya dapat dikembangkan

Model Incremental Development



Jenis-Jenis SDLC (2)

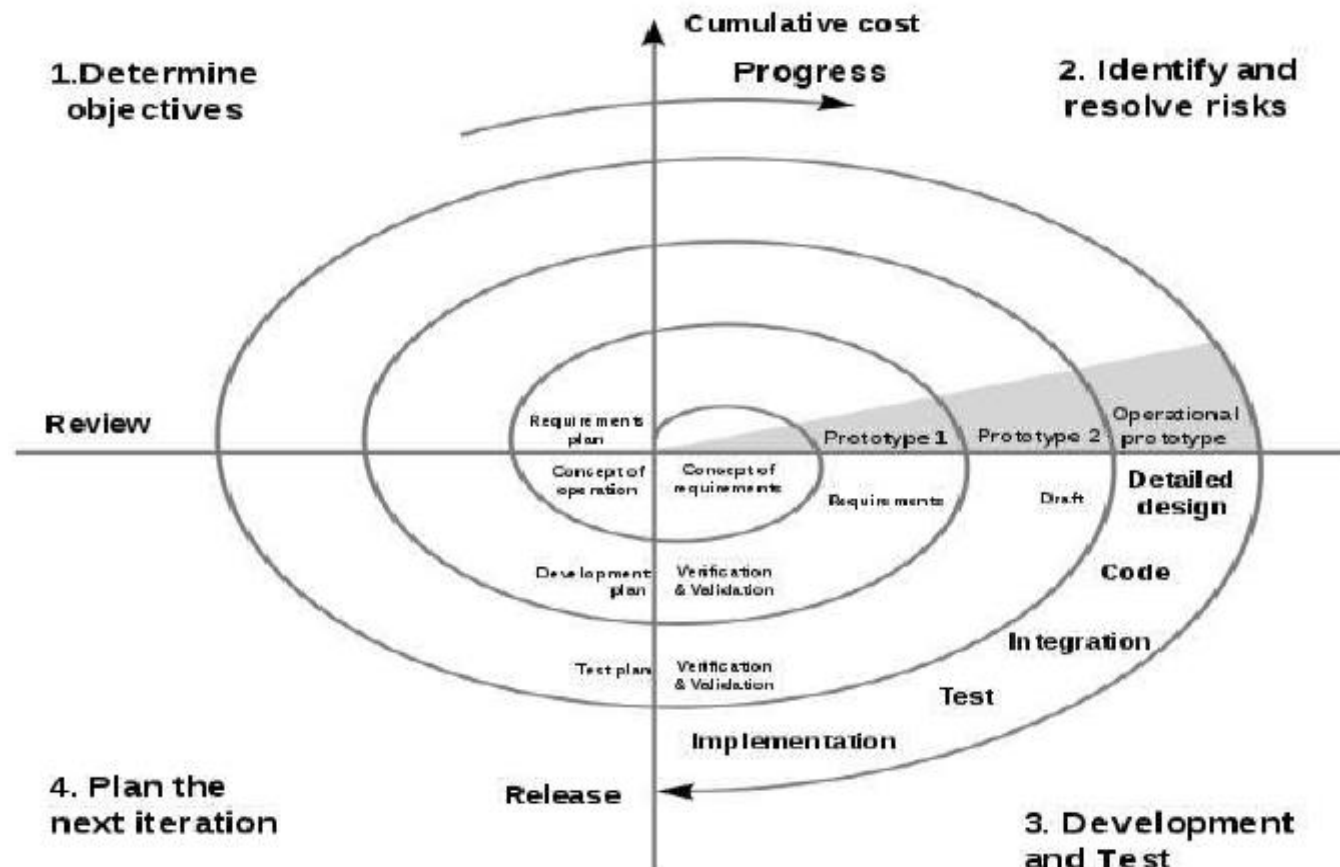
Model Spiral

- Merupakan kombinasi antara model waterfall dan prototipe
- Pemberlakukan analisa risiko pada tiap tahap.
- Terdapat 4 quadran pokok dalam pembangunan software
- Dokumentasi dibuat pada saat progress

4 Quadran pada model spiral

1. Mendefinisikan tujuan dan alternatif
2. Evaluasi alternatif; identifikasi dan pemecahan risiko
3. Development dan testing
4. Rencana untuk fase berikutnya

Jenis-Jenis SDLC - Model Spiral



Sumber: [http://en.wikipedia.org/wiki/File:Spiral_model_\(Boehm,_1988\).svg](http://en.wikipedia.org/wiki/File:Spiral_model_(Boehm,_1988).svg)

Jenis-Jenis SDLC (3)

Model Prototype

- Merupakan sketsa produk/software yang sederhana
- Memerlukan kontribusi yang besar dari stakeholder
- Model ini digunakan bagi konsumen yang memiliki keinginan, namun sulit menggambarkan apa yang sebenarnya dibutuhkan.

Fase prototype:

1. identifikasi kebutuhan - keinginan user
2. desain prototipe sederhana
3. verifikasi dan validasi (user)
4. implementasi
5. rilis

Jenis-Jenis SDLC (4)

Agile Model

Model pengembangan perangkat lunak yang menekankan pada kelincahan tim dalam menyelesaikan proyek tersebut.

Dalam pengembangannya, tim secara intensif berkolaborasi dengan konsumen. Contoh:

1. eXtreme Programming (XP)
2. Scrum

Jenis-Jenis SDLC (5)

Extreme Programming (XP) memiliki 4 fase:

- Planning

- Membuat user story (proses bisnis)

- Menentukan hal2 yang dibutuhkan

- Designing

- menggunakan kartu CRC (class responsibility collaborator)

- Coding

- pair programming, refactoring

- Testing

- unit testing

- customer testing

Jenis-Jenis SDLC (6)

Scrum memiliki beberapa ciri khas (Pressman, 2005):

- Planning
- Backlogging (membagi proyek menjadi beberapa bagian)
- Sprint (menyelesaikan masing2 backlog dengan dipecah menjadi beberapa satuan waktu-- umumnya 30 hari)
- Meeting, mencakup beberapa pertanyaan:
 - Apa yang kalian lakukan sejak pertemuan sebelumnya?
 - Hambatan apa saja yang kalian temui?
 - Apa rencana kalian untuk pertemuan berikutnya?
- Demonstrating to customer

Jenis-Jenis SDLC (7)

UP (Unified Process)

Suatu pendekatan dalam pengembangan perangkat lunak yang merupakan bentuk kesatuan proses pengembangan perangkat lunak itu sendiri, dari sisi fase/tahapan dan workflow.

UP umumnya digunakan pada proses-proses yang bersifat unified (terintegrasi), seperti UML.

Jenis-Jenis SDLC (8)

4 Fase UP (Pressman, 2005):

1. Inception

→ rencana proyek, model bisnis, dll

2. Elaboration

→ analisa kebutuhan, risiko, model, dll

3. Construction

→ model, code, test, manual (dokumen)

4. Transition

→ beta testing, migrasi sistem, feedback

Latihan..

1. Bentuk kelompok maks. 3 orang (1 leader)
2. Tentukan tema tugas besar yang telah ditentukan: Blog, Wiki, SCM, SIA (KRS/Rapor), Perpustakaan, E-Commerce, Project gallery, Social network!
3. Diskusikan model pengembangan mana yang paling tepat dan beri analisisnya, mengapa harus memilih model pengembangan tersebut!
4. Tentukan berapa pelaku yang terlibat di dalam sistem!
5. tentukan berapa proses bisnis yang ada!

Bacaan lanjut..

1. IBM, Rational Unified Process Best Practice.
http://www.ibm.com/developerworks/rational/library/content/03July/1000/1251/1251_bestpractices_TP026B.pdf
2. Ambler, Scott W. 2005. A Manager's Introduction to The Rational Unified Process (RUP).
<http://www.ambysoft.com/downloads/managersIntroToRUP.pdf>
3. Pressman, Roger S., Software Engineering : A Practitioner's Approach, 6th edition, McGraw-Hill International, 2005
- 4 Jurnal-jurnal lain yang mendukung