

Rekayasa Perangkat Lunak (Software Engineering)

Siti Sa'uda, M.Kom.

Software Engineering: A Practitioner's Approach, 6/e

Chapter 4

Pengembangan PL

Manifesto bagi Pengembangan PL yang Tangguh

“We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:

- *Individuals and interactions* over processes and tools
- *Working software* over comprehensive documentation
- *Customer collaboration* over contract negotiation
- *Responding to change* over following a plan

That is, while there is value in the items on the right, we value the items on the left more.”

Kent Beck et al

Apakah yang dimaksud “Tangguh”?

- Respon Efektif (cepat dan adaptif) pada Perubahan
- Komunikasi Efektif terhadap semua stakeholders
- Melibatkan konsumen pada tim
- Mengorganisasi sebuah tim sehingga kinerjanya dapat dikendalikan

Menghasilkan ...

- PL yang secara cepat dan bertahap disajikan.

Proses yang Tangguh

- Diarahkan oleh gambaran konsumen terhadap apa yang dibutuhkan (skenario)
- Menyadari bahwa rencana biasanya berumur pendek
- Mengembangkan PL secara iteratif dengan tekanan pada aktivitas konstruksi
- Menyajikan pertumbuhan PL jamak
- Beradaptasi seiring dengan perubahan yang terjadi

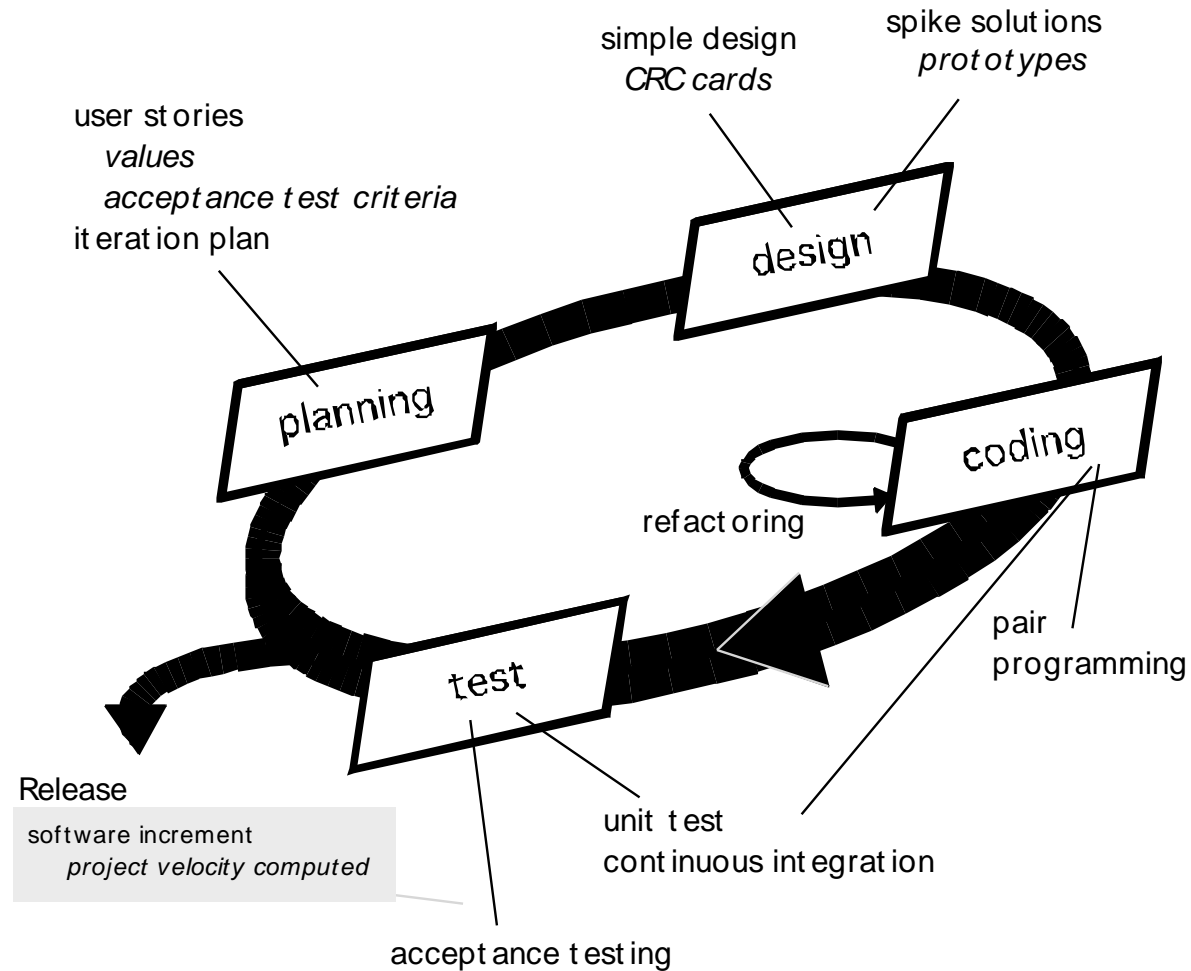
Extreme Programming (XP)

- Proses tangguh yang paling luas digunakan, yang dipelopori oleh Kent Beck
- Perencanaan XP
 - Mulai dari “keinginan user”
 - Tim memeriksa setiap keinginan dan menyebutkan harga
 - Keinginan2x tersebut dikelompokkan untuk proses penyelesaian yang bertahap
 - Komitmen dibuat pada tanggal penyajian
 - Setelah tahap pertama, “kecepatan proyek” digunakan untuk membantu menentukan tanggal berikutnya bagi tahapan yang lain

Extreme Programming (XP)

- XP Design
 - Mengikuti prinsip KIS
 - Mendorong penggunaan kartu CRC
 - Untuk permasalahan desain yang sukar, menyarankan penggunaan “spike solutions” sebuah desain prototipe
 - Mendorong “**refactoring**”—sebuah perbaikan iteratif terhadap desain program internal
- XP Coding
 - Merekomendasikan konstruksi tes unit sebelum coding dimulai
 - Mendorong “**pair programming**”
- XP Testing
 - Semua tes unit dieksekusi setiap hari
 - “**tes penerimaan**” ditentukan oleh konsumen dan dieksekusi untuk melihat fungsionalitas konsumen nyata

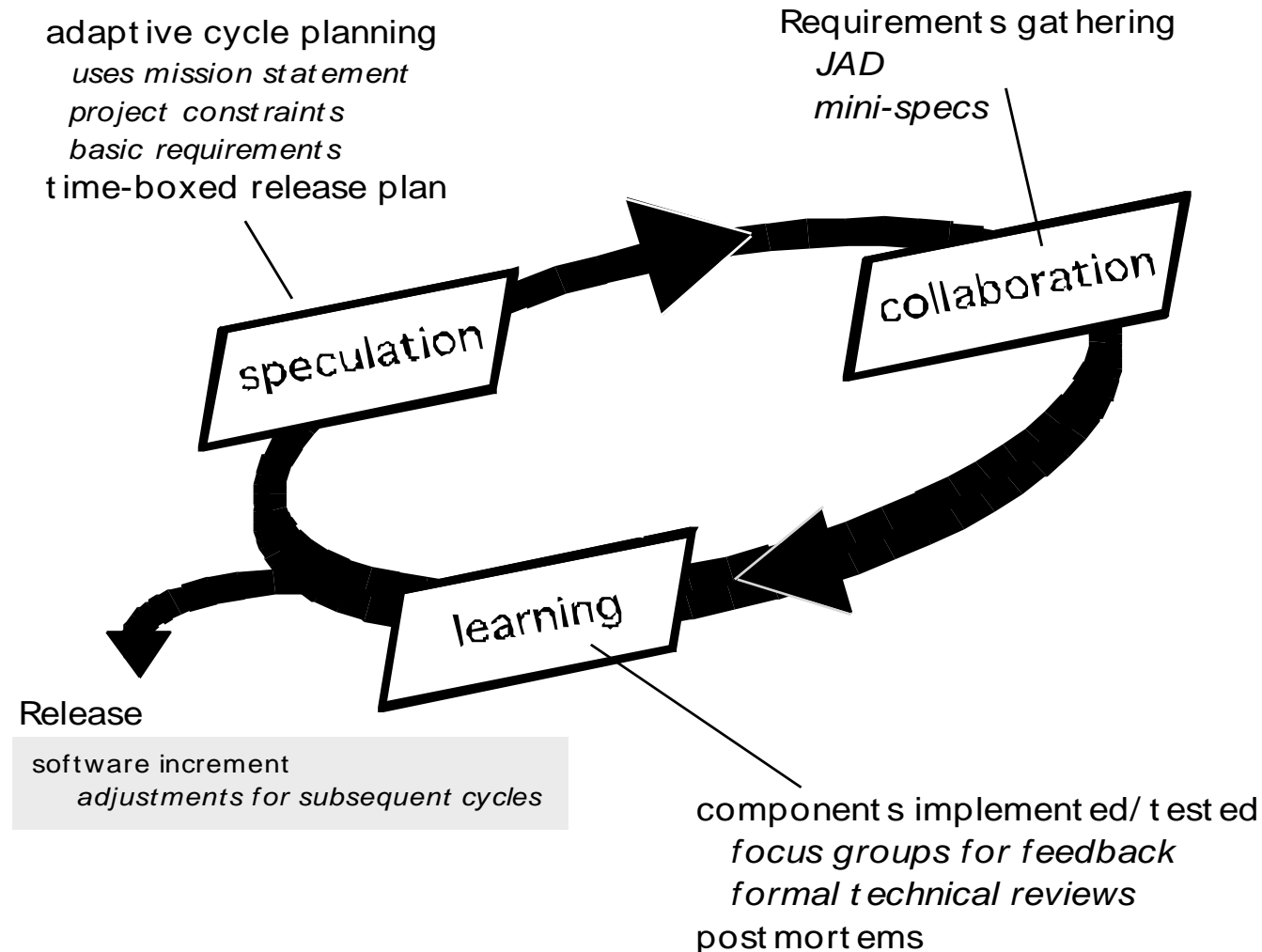
Extreme Programming (XP)



Pengembangan PL Adaptif

- Dipelopori oleh Jim Highsmith
- ASD — karakter yang membedakan
 - Perencanaan berbasis misi
 - Fokus berbasis komponen
 - Menggunakan “time-boxing”
 - Konsideran eksplisit pada resiko
 - Menekankan kolaborasi bagi pengumpulan kebutuhan
 - Menekankan pembelajaran melalui proses

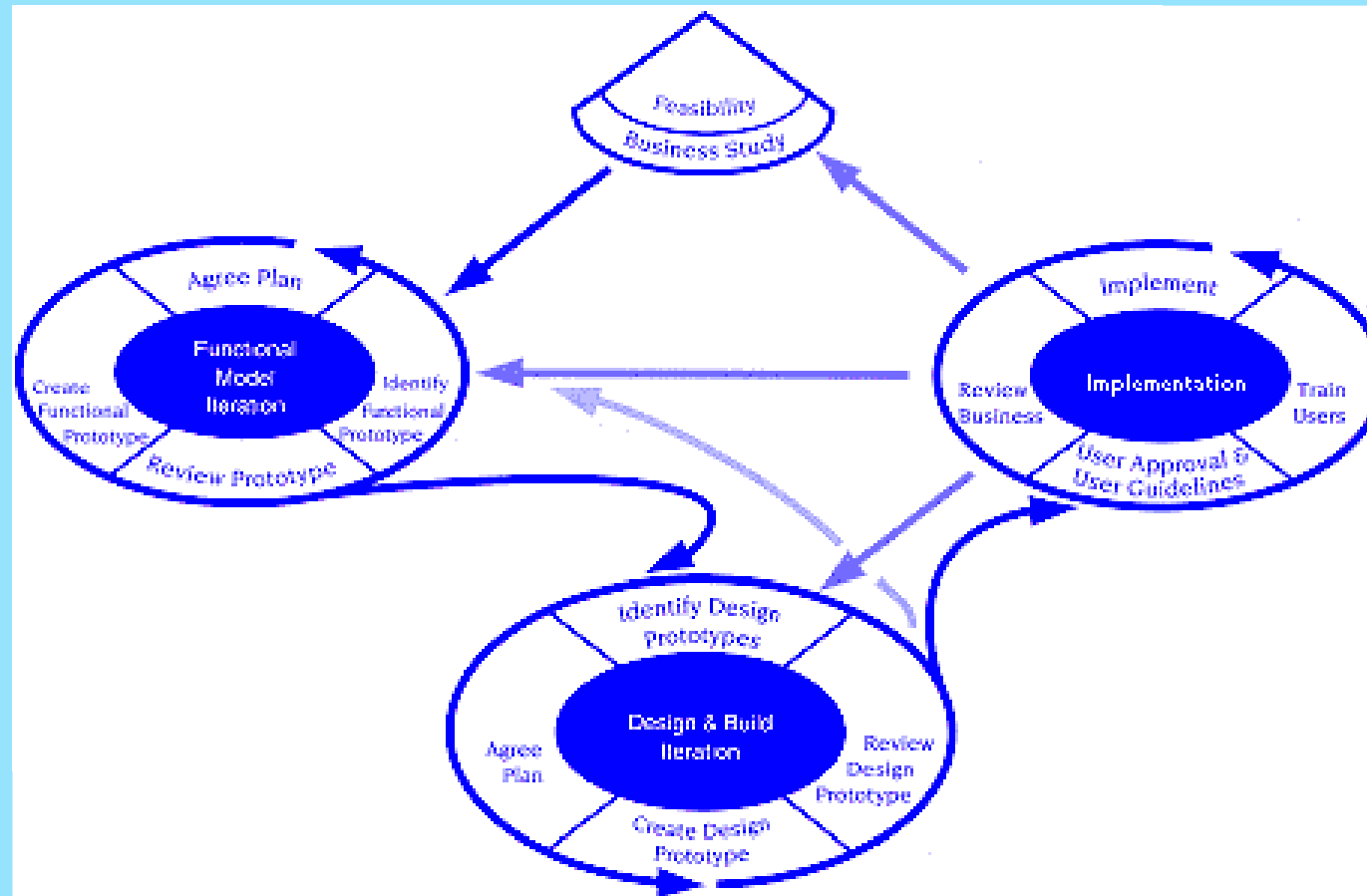
Adaptive Software Development



Metode Pengembangan Sistem Dinamis (Dynamic Systems Development Method)

- Dipromosikan oleh konsorsium DSDM (www.dsdm.org)
- DSDM—karakter yang membedakan
 - Mirip dalam banyak dengan XP dan/atau ASD
 - Sembilan prinsip-prinsip panduan :
 - Pelibatan user secara aktif adalah keharusan.
 - Tim DSDM harus diberdayakan untuk mengambil keputusan.
 - Fokus pada penyajian produk sesering mungkin.
 - Penerimaan dari tujuan bisnis adalah kriteria esensial untuk penerimaan penyajian.
 - Pengembangan bertahap dan berulang dibutuhkan untuk fokus pada solusi bisnis yang akurat.
 - Semua perubahan selaman pengembangan dapat dibalik.
 - Kebutuhan adalah dasar pada level tinggi
 - Pengujian terintegrasi dalam siklus kehidupan.

Dynamic Systems Development Method

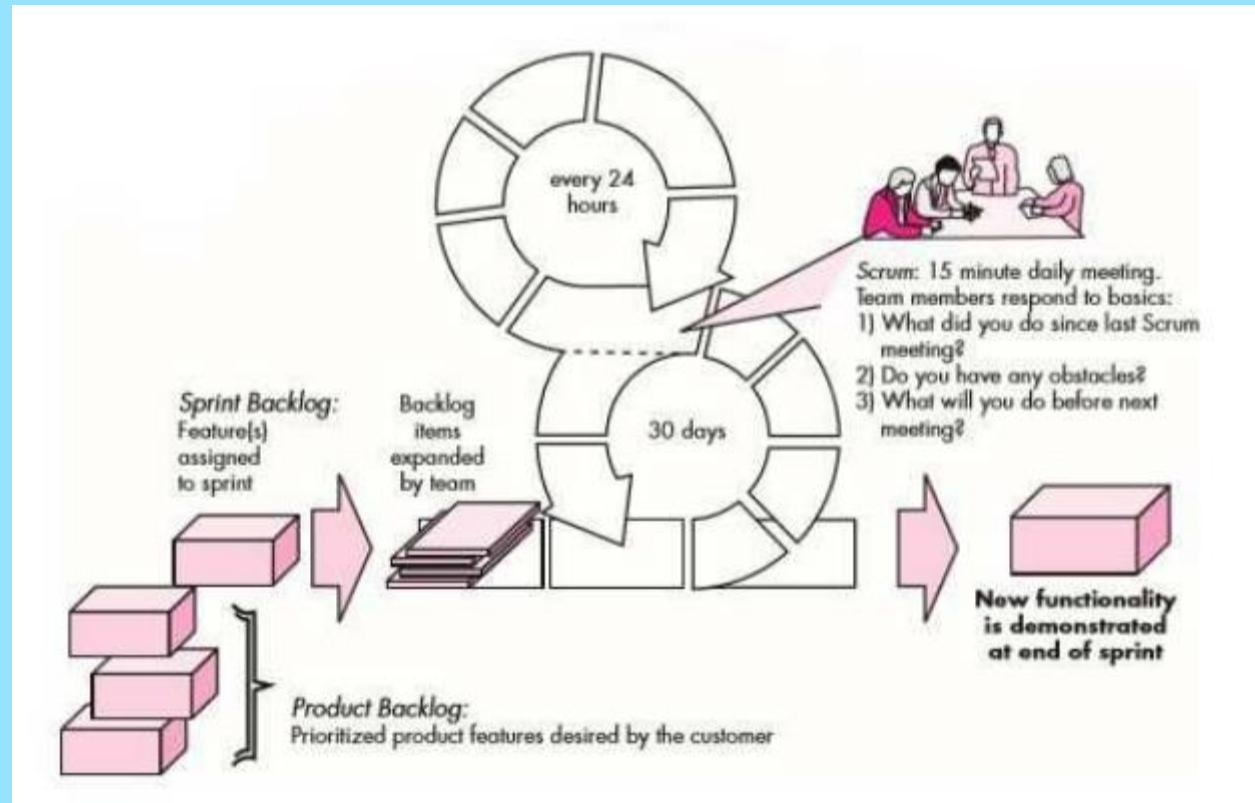


DSDM Life Cycle (with permission of the DSDM consortium)

Scrum

- Diusulkan oleh Schwaber dan Beedle
- Scrum—Karakter yang membedakan
 - Kerja pengembangan dipartisi menjadi “paket”
 - Pengujian dan dokumentasi berjalan seiring dengan konstruksi produk
 - Kerja terjadi dalam “Sprint” dan diturunkan dari “backlog” kebutuhan yang ada
 - Pertemuan sangat pendek dan beberapa kali diadalah tanpa kursi
 - “Demo” ditunjukkan pada konsumen dengan alokasi time-box

Scrum



Proses Scrum menekankan pada penggunaan serangkaian pola-pola pengerjaan (Noyes,2002) yang telah terbukti secara efektif pada proyek yang mempunyai jadwal yang ketat, *requirement* yang berubah-ubah, serta kebutuhan bisnis yang tinggi.

Masing-masing pola pengerjaan tersebut mengandung serangkaian aktivitas sebagai berikut:

- *Backlog*-merupakan daftar prioritas dari *requirement* pada proyek tersebut atau fitur-fitur yang menyediakan nilai-nilai bisnis untuk pelanggan. Hal yang membuat Scrum fleksibel adalah item-item dapat dimasukkan ke dalam *backlog* Manajer proyek akan mengevaluasi *blacklog* untuk kemudian memperbarui daftar prioritas sesuai kebutuhan.

- *Sprints*-terdiri atas unit-unit kerja yang dibutuhkan untuk mencapai *requirements* yang telah dibuat pada *backlog* dalam waktu yang ditentukan oleh *time-box* (biasanya 30 hari). Perubahan pada *backlog* tidak diperlihatkan pada tahap *sprints* oleh karena itu *sprints* memungkinkan anggota tim untuk bekerja dalam waktu yang singkat namun dengan lingkungan yang stabil.
- *Scrum meetings*-dapat disebut sebagai “rapat singkat” karena dilakukan dalam waktu 15 menit setiap harinya oleh anggota Scrum. Ada tiga pertanyaan yang harus dijawab oleh semua anggota tim (Noyes, 2002):
 - Apa yang sudah anda lakukan sejak pertemuan rapat terakhir?
 - Apa hambatan yang anda temui?
 - Apa yang ingin anda capai sebelum rapat tim berikutnya?

Rapat Scrum dipimpin oleh seorang *Scrum Master* yang memimpin rapat serta mengevaluasi tanggapan dari masing-masing anggota tim. Rapat Scrum memungkinkan anggota tim untuk menemukan hal-hal yang berpotensi menjadi masalah sedini mungkin.

- *Demos*-merupakan penyerahan hasil kerja secara bertingkat sehingga produk dapat didemonstrasikan dan dievaluasi bersama.

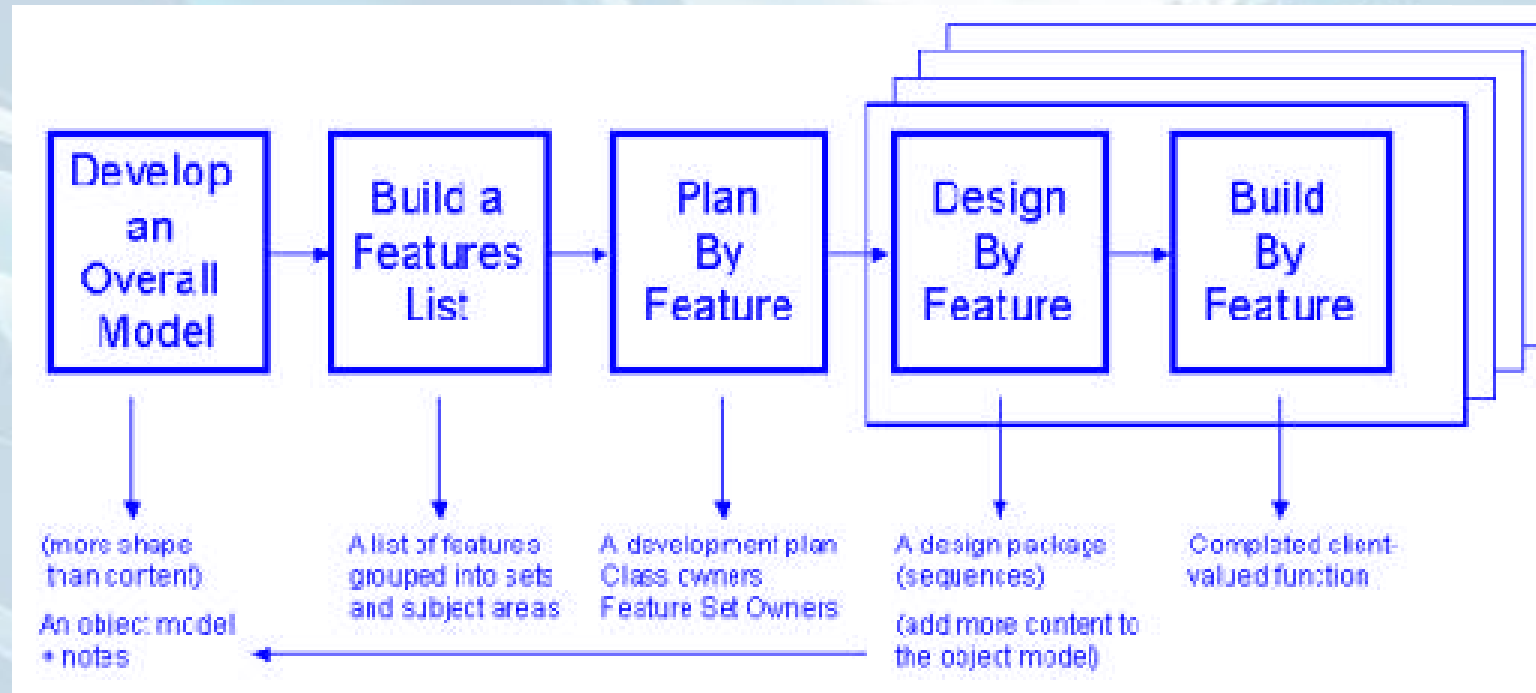
Crystal

- Diusulkan Cockburn dan Highsmith
- Crystal—karakter yang membedakan
 - Secara aktual sebuah model proses keluarga yang memungkinkan manuver berdasar karakteristik permasalahan
 - Komunikasi tatap muka ditekankan
 - Menyarankan penggunaan workshop refleksi untuk review kebiasaan kerja tim

Feature Driven Development

- Diusulkan oleh Peter Coad et al
- FDD—karakter yang membedakan
 - Penekanan pada definisi “features”
 - a *feature* “is a client-valued function that can be implemented in two weeks or less.”
 - Menggunakan template feature
 - <action> the <result> <by | for | of | to> a(n) <object>
 - Daftar feature dibuat dan “perencanaan berdasar “feature” dilakukan
 - Desain dan konstruksi bergabung dalam FDD

Feature Driven Development



Reprinted with permission of Peter Coad

Agile Modeling

- Diusulkan oleh Scott Ambler
- Menyarankan prinsip2x agile modeling
 - Model dengan sebuah tujuan
 - Menggunakan banyak model
 - Isi lebih penting dari representasi
 - Mengetahui model dan tool yang digunakan untuk membuatnya
 - Beradaptasi secara lokal