SDLC

Software Development Life Cycle

Fajar Pradana S.ST., M.Eng

FILKOM

SDLC Model

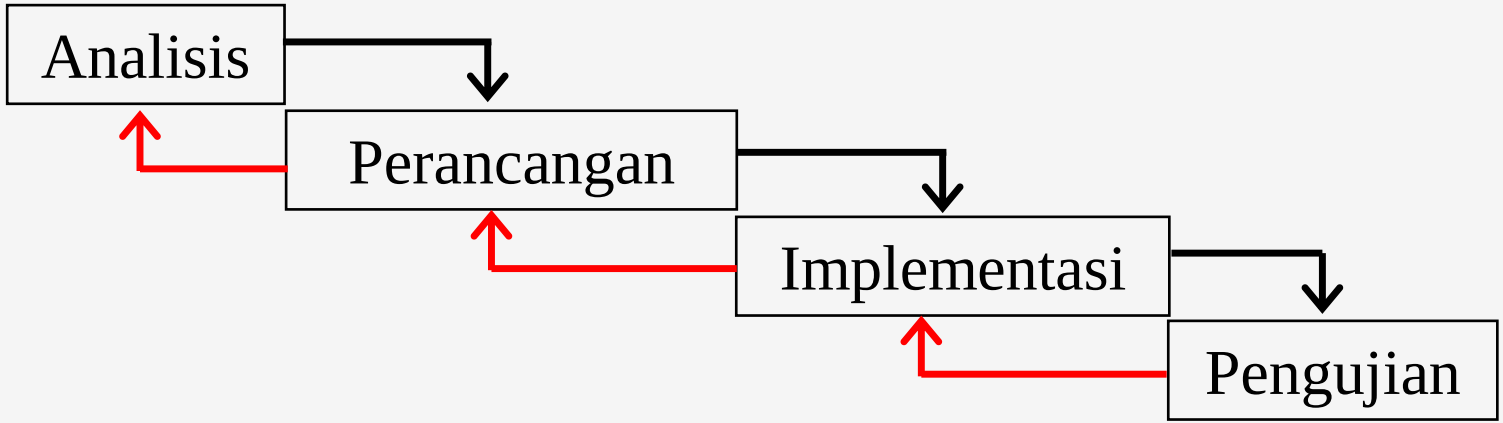
Sebuah framework yang mendeskripsikan performa aktivitas dari setiap stage dari pengembangan perangkat lunak.

01

Linear Sequential

Model pengembangan classic, yang tergolong SDLC jenis ini adalah Waterfall dan V-Shaped

Model Waterfall/Classic/Linear Sequential



Waterfall Strength

- Mudah dipahami, mudah untuk digunakan
- Menyediakan struktur untuk staff yang tidak berpengalaman
- Milestones dipahami dengan baik
- Requirement akan menjadi stabil
- Baik untuk pengendalian manajemen (plan, staff, track)
- “Works well” jika kualitas lebih penting daripada biaya atau jadwal

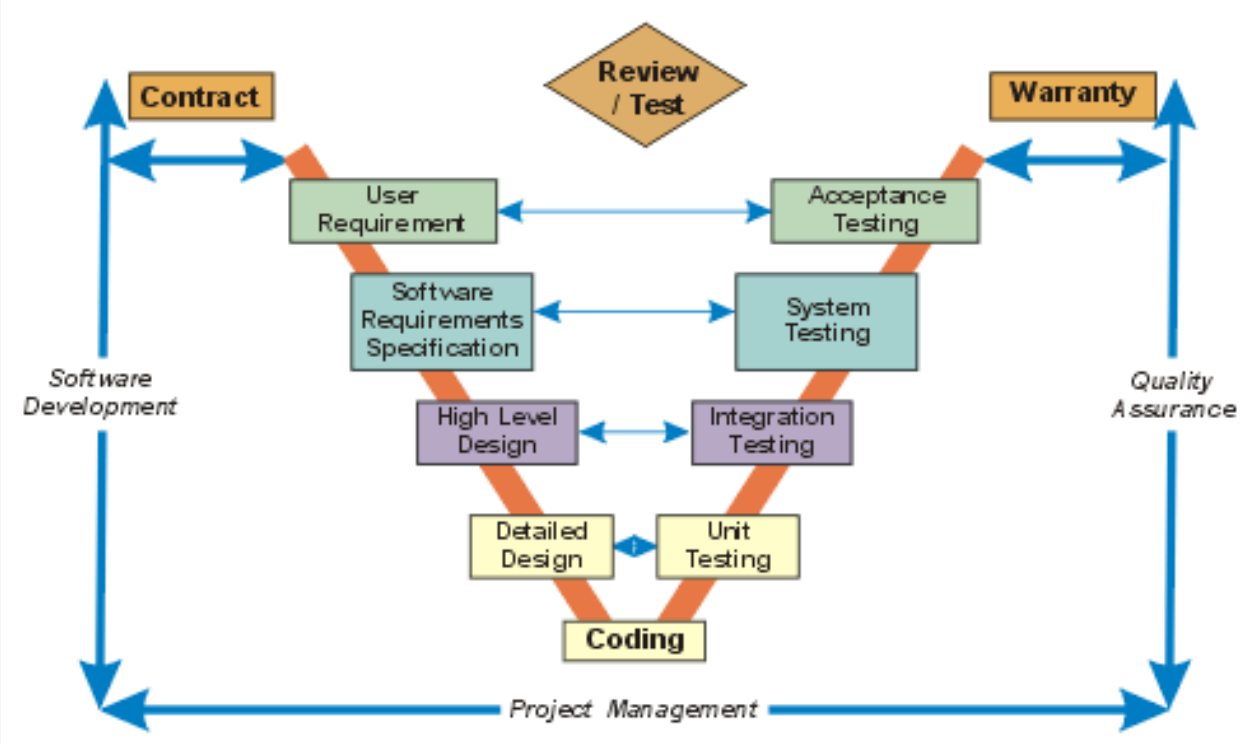
Waterfall Deficiencies

- Semua requirement harus diketahui di awal
- Deliverables dibuat untuk setiap fase dianggap “freeze”- menghambat fleksibilitas
- Tidak mencerminkan sifat pemecahan masalah pengembangan perangkat lunak - iterasi dari fase
- Integrasi adalah salah satu big bang di akhir
- Sedikit kesempatan bagi pelanggan untuk melihat sistem (sampai mungkin terlalu terlambat)

When Use Waterfall

- Requirements are very well known
- Product definition is stable
- Technology is understood
- New version of an existing product
- Porting an existing product to a new platform.

Model V


Review / Test
Contract
Warranty
User Requirement
Acceptance Testing
Software Requirements Specification
System Testing
High Level Design
Integration Testing
Detailed Design
Unit Testing
Coding
Software Development
Quality Assurance
Project Management

V-Shaped Steps

- Project and Requirements Planning – allocate resources
- Product Requirements and Specification Analysis – complete specification of the software system
- Architecture or High-Level Design – defines how software functions fulfill the design
- Detailed Design – develop algorithms for each architectural component
- Production, operation and maintenance – provide for enhancement and corrections
- System and acceptance testing – check the entire software system in its environment
- Integration and Testing – check that modules interconnect correctly
- Unit testing – check that each module acts as expected
- Coding – transform algorithms into software

V-Shaped Strengths

- Menekankan perencangan untuk verifikasi dan validasi produk difase yang lebih awal.
- Setiap delivery harus melewati proses testing
- Manajemen project dapat melakukan tracking dari milestone
- Easy to use

V-Shaped Weaknesses

- Does not easily handle concurrent events
- Does not handle iterations or phases
- Does not easily handle dynamic changes in requirements
- Does not contain risk analysis activities

When to use the V-Shaped Model

- Excellent choice for systems requiring high reliability – hospital patient control applications
- All requirements are known up-front
- When it can be modified to handle changing requirements beyond analysis phase
- Solution and technology are known

02

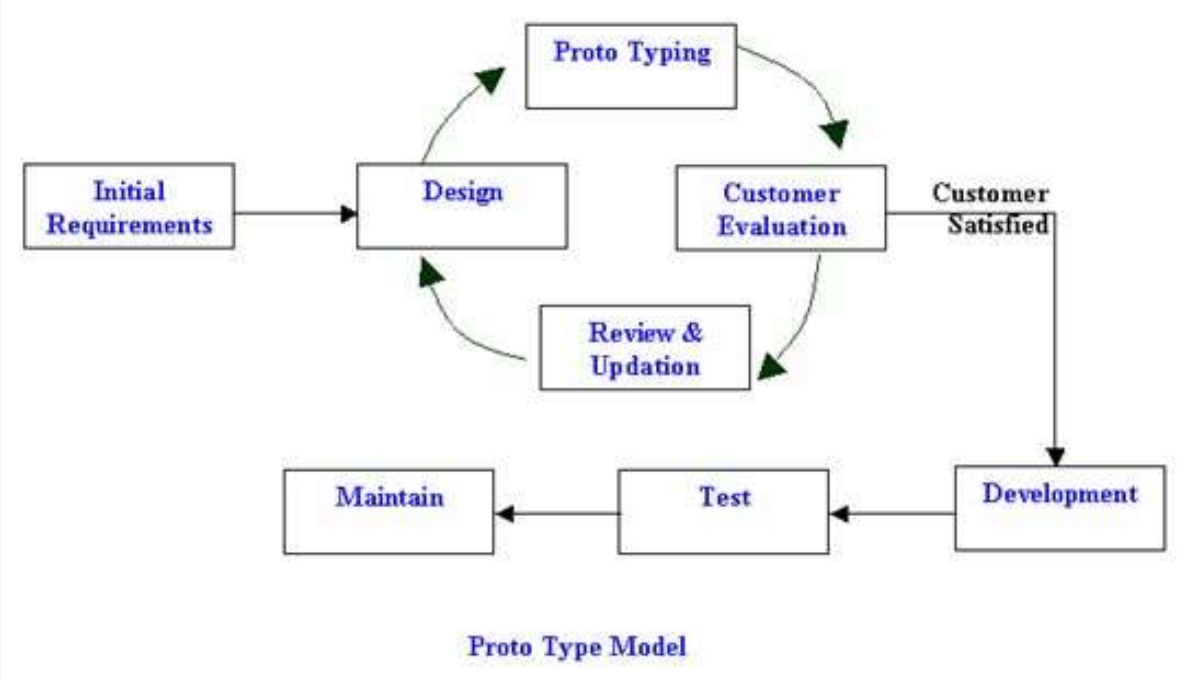
Iterative Model

Salah satu cirinya terdapat kegiatan yang berulang. Prototyping dan Spiral termasuk dalam jenis SDLC ini.

Prototyping Model

- Developers build a prototype during the requirements phase
- Prototype is evaluated by end users
- Users give corrective feedback
- Developers further refine the prototype
- When the user is satisfied, the prototype code is brought up to the standards needed for a final product.

Prototyping SDLC



Structured Evolutionary Prototyping Steps

- A preliminary project plan is developed
- An partial high-level paper model is created
- The model is source for a partial requirements specification
- A prototype is built with basic and critical attributes
- The designer builds
    - the database
    - user interface
    - algorithmic functions
- The designer demonstrates the prototype, the user evaluates for problems and suggests improvements.
- This loop continues until the user is satisfied

Structured Evolutionary Prototyping Strengths

- Customers can “see” the system requirements as they are being gathered
- Developers learn from customers
- A more accurate end product
- Unexpected requirements accommodated
- Allows for flexible design and development
- Steady, visible signs of progress produced
- Interaction with the prototype stimulates awareness of additional needed functionality

Prototyping Weaknesses

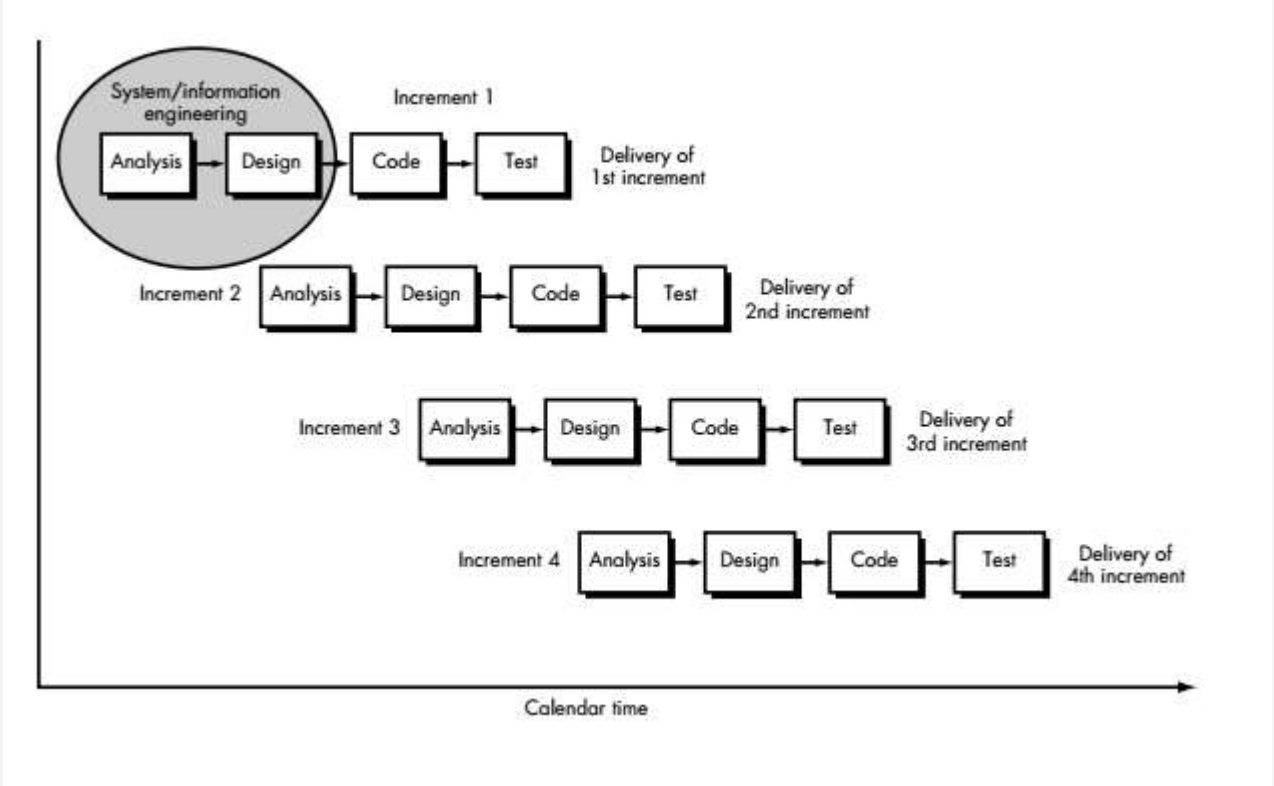
- Tendency to abandon structured program development for “code-and-fix” development
- Bad reputation for “quick-and-dirty” methods
- Overall maintainability may be overlooked
- The customer may want the prototype delivered.
- Process may continue forever (scope creep)

03

Incremental Model

Model pengembangan secara bertahap (incremental). Kombinasi linear skuensial model dan filosofi iterative dari prototyping

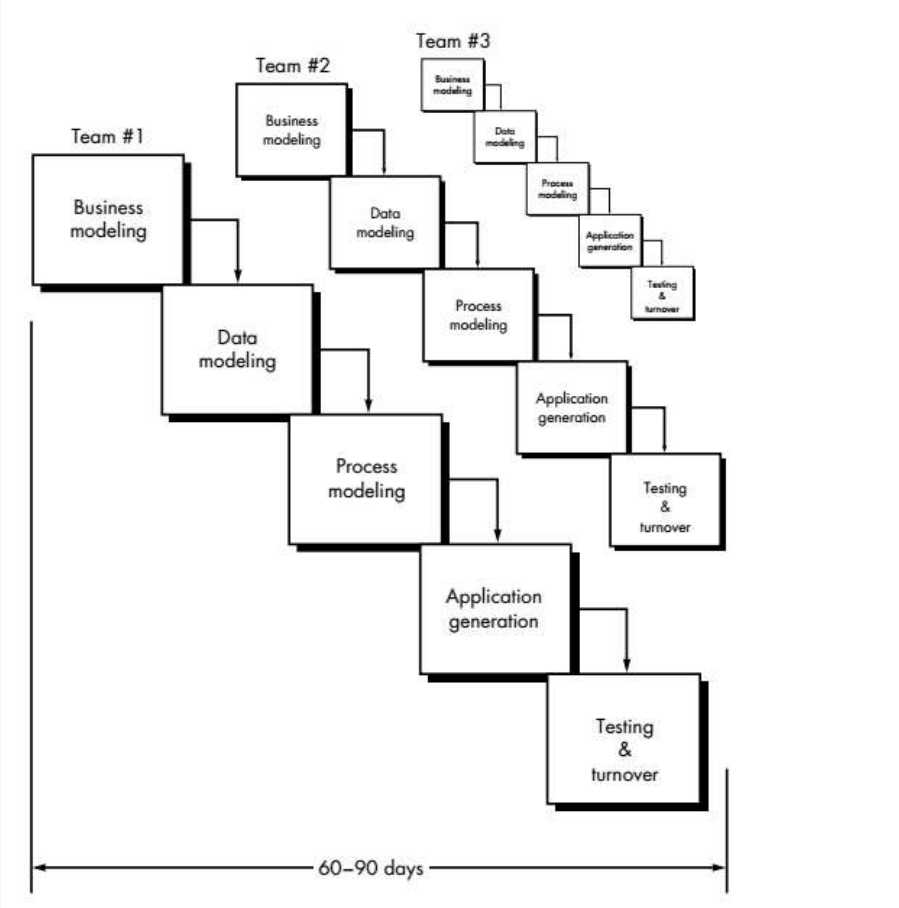
Incremental Model



Rapid Application Model (RAD)

- Requirements planning phase (a workshop utilizing structured discussion of business problems)
- User description phase – automated tools capture information from users
- Construction phase – productivity tools, such as code generators, screen generators, etc. inside a time-box. (“Do until done”)
- Cutover phase -- installation of the system, user acceptance testing and user training

RAD Model


Team #1
Business modeling
Data modeling
Process modeling
Application generation
Testing & turnover
Team #2
Business modeling
Data modeling
Process modeling
Application generation
Testing & turnover
Team #3
Business modeling
Data modeling
Process modeling
Application generation
Testing & turnover
60–90 days

RAD Strengths

- Reduced cycle time and improved productivity with fewer people means lower costs
- Time-box approach mitigates cost and schedule risk
- Customer involved throughout the complete cycle minimizes risk of not achieving customer satisfaction and business needs
- Uses modeling concepts to capture information about business, data, and processes.

RAD Weaknesses

Accelerated development process must give quick responses to the user

Risk of never achieving closure

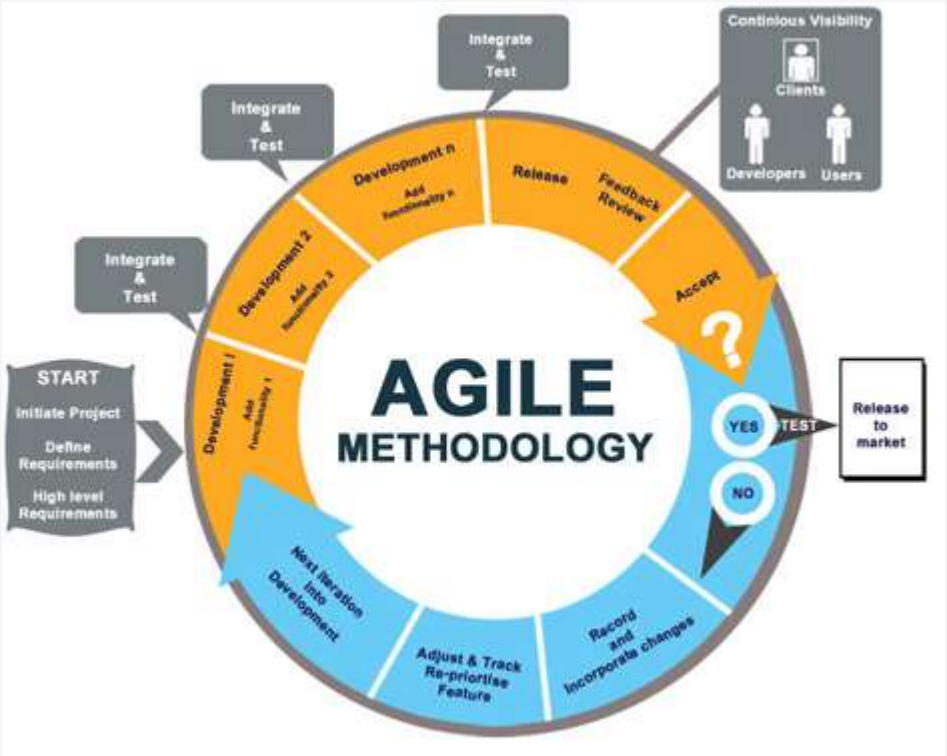
Hard to use with legacy systems

Requires a system that can be modularized

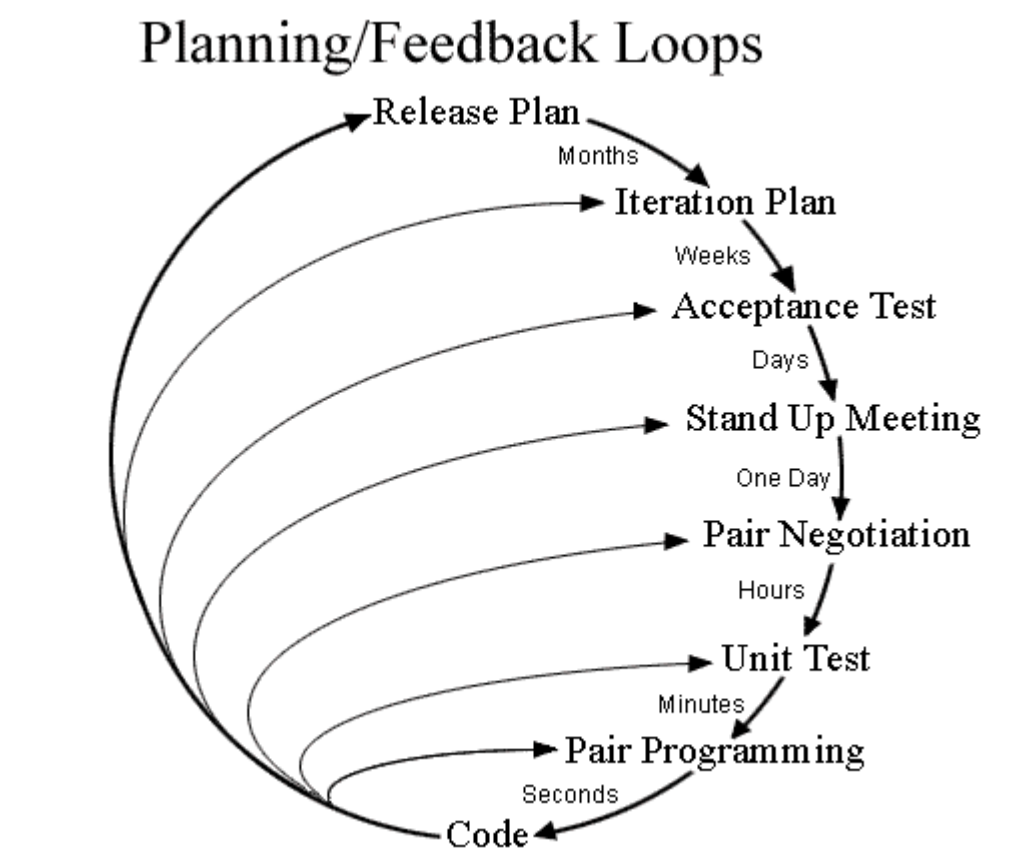
Developers and customers must be committed to rapid-fire activities in an abbreviated time frame.

04 Other SDLC

Agile Methodology



Extreme Programming



Terima Kasih

Ada Pertanyaan