**Contents**

1. Purpose 5

2. Scope 5

3. Definition 5

4. Guidelines 5

4.1. V-shaped model 5

4.1.1. Overview 5

4.1.2. When to use 5

4.1.3. Life cycle 5

4.1.4. Activities Description 6

4.1.5. Advantage 6

4.1.6. Disadvantage 7

4.2. Incremental Model 7

4.2.1. Overview 7

4.2.2. When to use 7

4.2.3. Life cycle 7

4.2.4. Activities Description 7

4.2.5. Advantage 9

4.2.6. Disadvantage 9

4.3. Prototyping Model 9

4.3.1. Overview 9

4.3.2. When to use 9

4.3.3. Life cycle 9

4.3.4. Activities Description 10

4.3.5. Advantage 12

4.3.6. Disadvantage 12

# Purpose

* To define software development life cycle models implemented in software projects.

# Scope

# Definition

* Software development life cycle: a framework that describes the activities performed at each stage of a software development project.

# Guidelines

The common software lifecycle models are:

* V-shaped model
* Incremental model
* Prototyping model

## V-shaped model

### Overview

* A variant of the Waterfall that emphasizes the verification and validation of the product. Each next phase in this model must beginning only after the previous phase is over.
* Testing of the product is planned in parallel with a corresponding phase of development.

### When to use

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

### Life cycle

Requirements Analysis

Architectural Design

Detailed Design

Implementation

Unit Testing

Integration Testing

System Testing

Unit test planning

Integration test planning

System test planning

### Activities Description

|  |  |  |  |
| --- | --- | --- | --- |
| **Process/Activity** | **Description** | **Input** | **Work product** |
| Project Launch | Allocate resources  Refer to Project Launch process definition | SOW | SDP-01, SDP-02 |
| Requirement Analysis | - Complete requirement specification of the software system  - Plan for system test  - Refer to Requirement Analysis Process Definition | SOW | RMT, SRS, STP |
| Architectural Design | - Defines how software functions fulfill the design  - Plan for integration test  - Refer to Architectural Design Process Definition | SRS | SAD, ITP |
| Detailed Design | - Develop algorithms for each architectural component  - Plan for unit test  - Refer to Detailed Design Process Definition | SAD | SDD, UTP |
| Implementation | Transform algorithms into software  - Refer to SW Implementation Process Definition | SDD | Source code |
| Unit test | Check that each module acts as expected  - Refer to Unit test Process Definition | Source code, UTP, SDD | UTS, Source code after UT |
| Integration test | Check that modules interconnect correctly  - Refer to Integration Test Process Definition | Source code after UT, ITP, SAD | ITS, Source code after IT |
| System Test | Check the entire software system in its environment  - Refer to System Test Process Definition | Source code after IT, STP, SRS | STS, final source code |
| Shipment | Shipment products to customer  - Refer to Shipment Process Definition | Agreement  All done products | Release Notes  Deliverable list  Deliverable package |
| Project Closing | Collect lesson learnt from project  - Refer to Project Closing Process Definition | All management documents | PJ Closing Report  Post Analysis Report  Defect Analysis Report |

### Advantage

* Emphasize planning for verification and validation of the product in early stages of product development
* Each deliverable must be testable
* Project management can track progress by milestones
* Easy to use

### Disadvantage

* Does not easily handle concurrent events
* Does not handle iterations or phases
* Does not easily handle dynamic changes in requirements
* Software is developed during the implementation phase, so no early prototypes of the software are produced.
* Model doesn't provide a clear path for problems found during testing phases.

## Incremental Model

### Overview

* Construct a partial implementation of a total system
* Then slowly add increased functionality
* The incremental model prioritizes requirements of the system and then implements them in groups.
* Each subsequent release of the system adds function to the previous release, until all designed functionality has been implemented.

### When to use

* Risk, funding, schedule, program complexity, or need for early realization of benefits.
* Most of the requirements are known up-front but are expected to evolve over time
* A need to get basic functionality to the market early
* On projects which have lengthy development schedules
* On a project with new technology

### Life cycle

Detailed Design

Implementation

V & V

Testing

Release

Increment - 1

Increment - 2

Detailed Design

V & V

Implementation

Testing

Release

Software Requirements Analysis

Architectural Design

Project Launch

### Activities Description

|  |  |  |  |
| --- | --- | --- | --- |
| **Process/ Activity** | **Description** | **Input** | **Work product** |
| Project Launch | Allocate resources  Refer to Project Launch process definition | SOW | SDP-01, SDP-02 |
| Requirement Analysis | - The first important steps in the incremental approach is as usual to define the requirements of the product. These first requirements should be basic and not very detailed. From these initial requirements increments are defined and the requirements are mapped to an increment which suits those requirements.  - Verification and validation in this phase is important. Requirement Specification need to be verified by V&V team and prensented before Higher Manager by holding DR meeting. This also be gotten the confirmation from customer.  - System test plan should be created as draft version  - Refer to Requirement Analysis Process Definition | SOW | RMT, SRS, STP |
| Architectural Design | - The system architecture for the whole system is designed. After this the incremental development cycle begins and continues until the system is finished.  - To be able to integrate the increments to a whole system the architecture of the system needs to be designed before the increments are started to be developed. The architecture needs to identify the interfaces between the increments and how they will be integrated.  - Plan initialization for integration test  - Refer to Architectural Design Process Definition | SRS | SAD, ITP |
| Detailed Design | - First the requirements for the increment is refined and is made more detailed. Second the increment is developed, tested and integrated into the system. Then the system is tested.  - While the increment is developed the requirements for the next increment can be refined and changed. It’s important though that the requirements for the increment being developed cannot be changed.  - When the increment is finally developed, tested and integrated it can be released to the customer. Meanwhile development will start on the next increment. When each increment has been developed it’s integrated with the previously developed increments. These increments, or subsystems, will finally be integrated to make the whole system.  - The verification of these results is made exactly the same way as in other models; all will be verified and validated by V&V team. The difference here is that you need to do technical documents reviewing and validation test for each of the increments released, this is because each increment has separate requirements. When each increment is integrated to the system it needs to go through integration testing and an validation test before being released.  - Depending on the duration of an incremental cycle, the DR for incremental cycle shall be done after finishing one incremental cycle or at least once every 1 month.  - DR for shipment shall be done before each official shipment following agreement. | SAD | SDD, UTP |
| Implementation | SDD | Source code, UTS |
| Testing | Source code after unit test | ITS, STS |
| Release | All technical documents,  Source code after integration test  ... | Release Notes  Deliverable list  Deliverable Package |
| Project Closing | Collect lessons learned from project  - Refer to Project Closing Process Definition | All management documents | PJ Closing Report  Post Analysis Report  Defect Analysis Report |

### Advantage

* Develop high-risk or major functions first
* Each release delivers an operational product
* Customer can respond to each build
* Uses “divide and conquer” breakdown of tasks
* Lowers initial delivery cost
* Initial product delivery is faster
* Customers get important functionality early
* Risk of changing requirements is reduced

### Disadvantage

* Requires good planning and design
* Requires early definition of a complete and fully functional system to allow for the definition of increments
* Well-defined module interfaces are required (some will be developed long before others)
* Total cost of the complete system is not lower

## Prototyping Model

### Overview

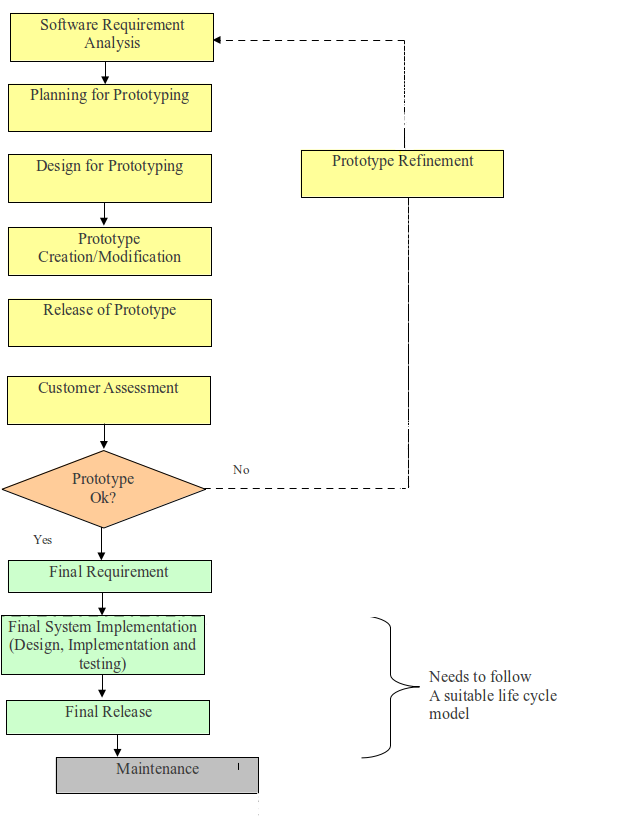
The Prototyping Model is a systems development method in which a prototype (an early approximation of a final system or product) is built, tested, and then reworked as necessary until an acceptable prototype is finally achieved from which the complete system or product can now be developed. This model works best in scenarios where not all of the project requirements are known in detail ahead of time. It is an iterative, trial-and-error process that takes place between the developers and the users. There are several steps in the 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.

### When to use

* Requirements are unstable or have to be clarified
* Develop user interfaces
* Short-lived demonstrations
* New, original development
* With the analysis and design portions of object-oriented development.

### Life cycle



### Activities Description

|  |  |  |  |
| --- | --- | --- | --- |
| **Process** | **Description** | **Input** | **Work product** |
| Project Launch | Allocate resources  Refer to Project Launch process definition | SOW | SDP-01, SDP-02 |
| Requirement Analysis | Similar to the Software Requirements analysis phase of the V-shaped Model, but not as comprehensive. The information collected is usually limited to a subset of the complete system requirements. | SOW | SRS draft |
| Planning for Prototyping | Prototyping must also be planned and controlled. Certain important parameters to be planned include :  - Breadth of functionality needed in the prototype at first and later on  - Completion criteria for the iteration cycle  - Composition of the team (users, developers, other stakeholders)  - Level of fidelity(dynamic) needed in the prototype at first, later on  - Maximum length of an iteration cycle  - Purpose of the prototype at first, later on  - Ways to manage conflict between team members, build consensus | SDP-01, SDP-02 (draft) | SDP-01, SDP-02 (updated) |
| Design for prototyping | Once the initial layer of requirements information is collected, or new information is gathered, it is rapidly integrated into a new or existing design so that it may be folded into the prototype. | SRS | SDD for prototype |
| Prototype Creation /Modification | The information from the design is rapidly rolled into a prototype. This may mean the creation/modification of paper information, new coding, or modifications to existing coding. | SDD | Source code for prototype, test design, test report |
| Release of Prototype | - PJ leader prepares the prototype release package after validation by V&V team.  - Depending on the duration of a prototyping cycle, the DR for prototyping shall be done after finishing one prototyping cycle or at least once every 1 month.  - DR for shipment shall be done before each official shipment following agreement. | All done prototypes | Prototype to be released, release notes, Final SRS |
| Customer Assessment | The prototype is presented to the customer for review. Comments and suggestions are collected from the customer. | All done prototype | Excel sheet to log the changes |
| Prototype Refinement | - The prototype is refined based on the information collected from the customer. The prototype is revised to make it more effective and efficient. | Excel sheet to log the changes | Enhanced source code of prototype (which will eventually become the product), review records, test design and report |
| Final System Implementation | Based on the purpose to be achieved, the prototype can be throwaway or evolutionary. Often, the prototype code is thrown away and entirely new programs are developed once requirements are identified. The final system is implemented using a suitable life cycle model, where more emphasis is given for documentation, verification and validation activities etc. | Final SRS, protype | Final SRS, SAD, SDD, review records, Traceability matrix, Source code, Test design and reports |
| Project Closing | Collect lesson learnt from project  - Refer to Project Closing Process Definition | All management documents | PJ Closing Report  Post Analysis Report  Defect Analysis Report |

### Advantage

* 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.

### Disadvantage

* 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)