

# Chap 2: Software Processes

---

## The software Process

A structured set of activities require to develop a software system

All involve:

- Specification
- Design and Implementation
- Validation
- Evolution

A software process model: An abstract presentation of a process

## Some software process models

The waterfall model:

- Plan-driven model
- Separate and distinct phases of specification and development

Incremental development

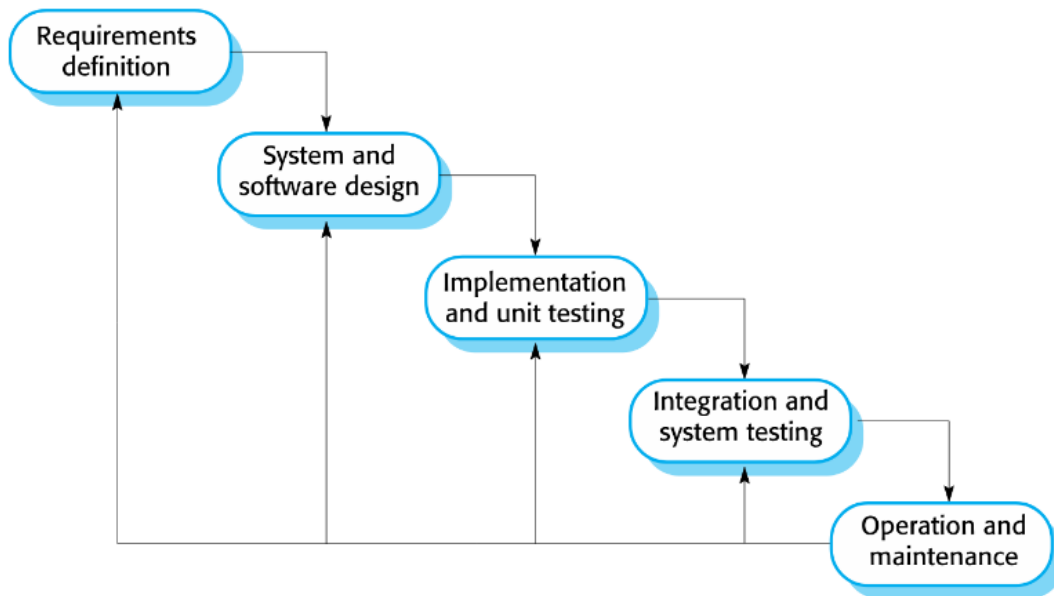
- Specification, development and validation are interleaved.
- May be plan-driven or agile

Integration and configuration

- The system is assemble from existing configurable components
- May be plan-driven or agile

*In practice, most large systems are developed using a process that incorporates elements from all these models.*

## the WATERFALL model



In principle, a phase has to be complete before moving onto the next phase.

The main drawback:

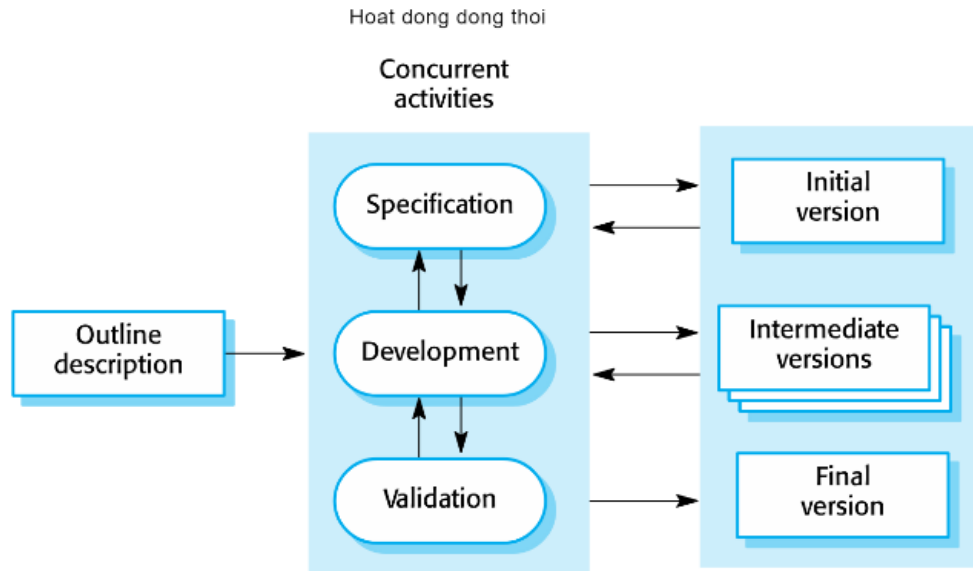
- difficulty of accommodating change after the process is underway

Mostly used for large systems engineering projects:

- Developed at several sites
- The plan-driven nature of the waterfall model helps coordinate the work

When the requirements are well-understood and changes will be fairly limited during the design process

## Increment Development



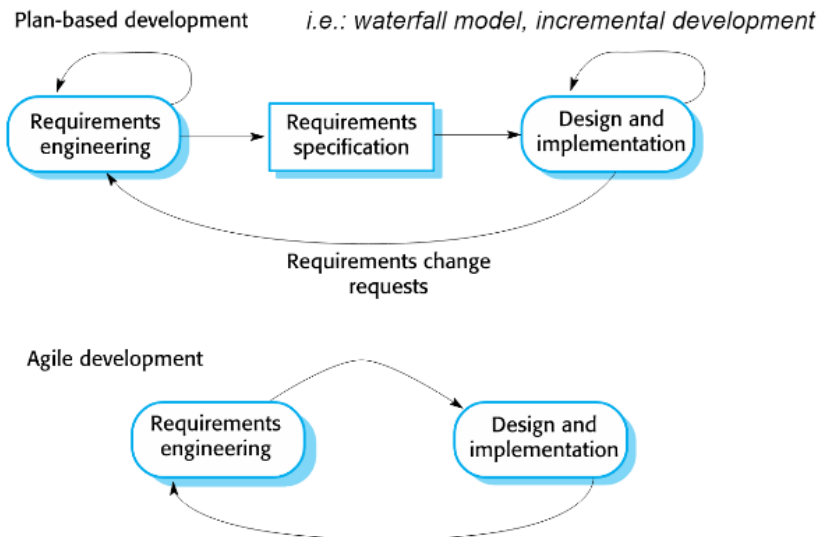
#### ID BENEFITS:

- Reduce the cost when the user want to change requirements
- Easier to get the feedback
- More rapid delivery and deployment of useful software

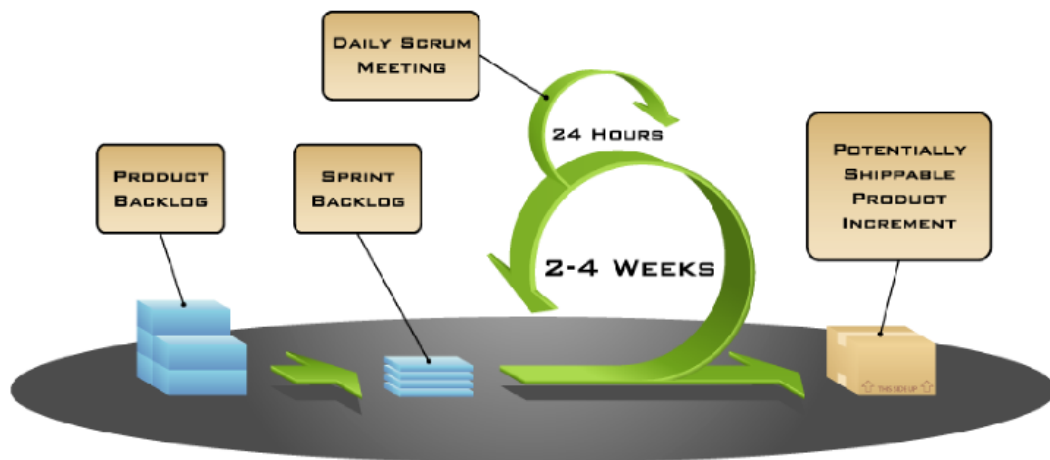
#### ID PROBLEMS:

- The process is not visible:
  - Managers need regular deliverables
  - Not cost-effective to produce documents for every version
- System structure tends to degrade as new increments are added
  - Need time and money on refactoring
  - Regular change tend to corrupt the structure
  - Incorporating further software changes becomes increasingly difficult and costly

## AGILE development



## SCRUM — THE MOST POPULAR AGILE DEVELOPMENT APPROACHES



PROs	CONs
More flexible	Hard to predict
Product get to market faster	Final product is not released first
Better communication	Documentation gets left behind

bat sua qa nhieu

moi version phai co tai lieu di kem de huong dan cho khach hang T^T qua lau

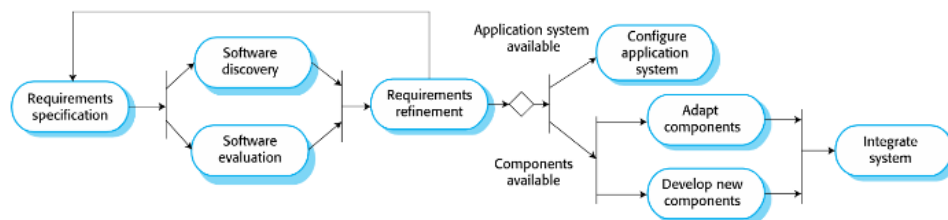
## Reused-oriented SOFTWARE engineering

Base on software reuse where systems are integrated from existing components or application system (COTS - Commercial-off-the-shelf) systems.

- Stand-alone application systems (COTS)
- Package objects/component framework such as .NET or J2EE
- Web services

Reused elements maybe configured to adapted their behaviour and functionality to a user's requirement

Reuse is now the standard approach for building many types of business system



## Advantages and disadvantages

Reduced costs and risk as less software is developed from scratch

Faster delivery and deployment

Requirements compromises are inevitable so may not meet real needs

Loss of control over evolution of reused part

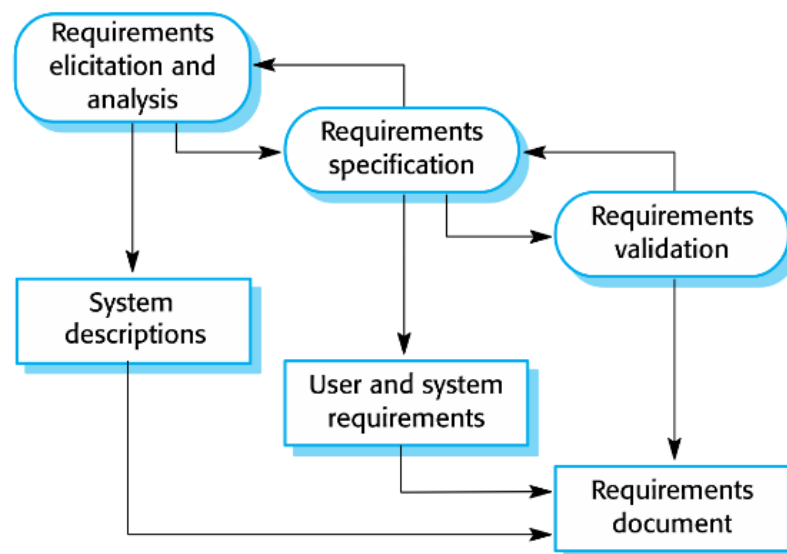
## PROCESS ACTIVITIES

### ACTIVITY: Software Specification

The process of establishing what services are required and the constraints on the system's operation and development

Use: Requirements engineering process

- Requirements elicitation and analysis
- Requirements specification
- Requirement validation



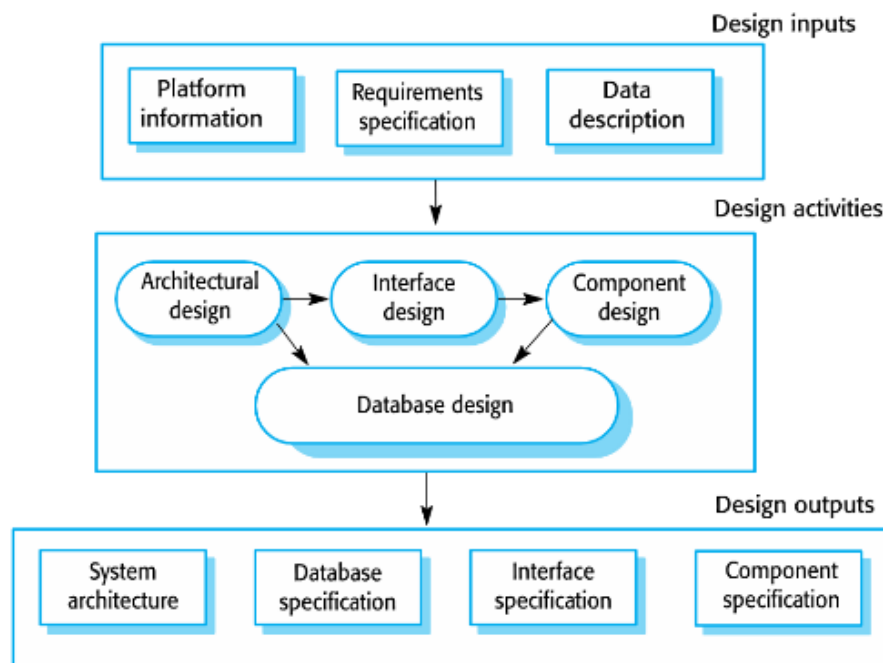
### ACTIVITY: Software design and implementation ~ software development

The process of converting the system specification into an executable system

Two (sub) activities:

- Software design: design structure
- Implementation: translate this structure into an executable program
- 2 activities are closely related and may be inter-leaved

## A general model of the design process



## System implementation

The software is implemented either by developing a program or programs or by configuring an application system

Design and implementation are interleaved activities for most types of software system

Programming is an individual activity with no standard process

Debugging is the activity of finding program faults and correcting

## ACTIVITY: Software validation

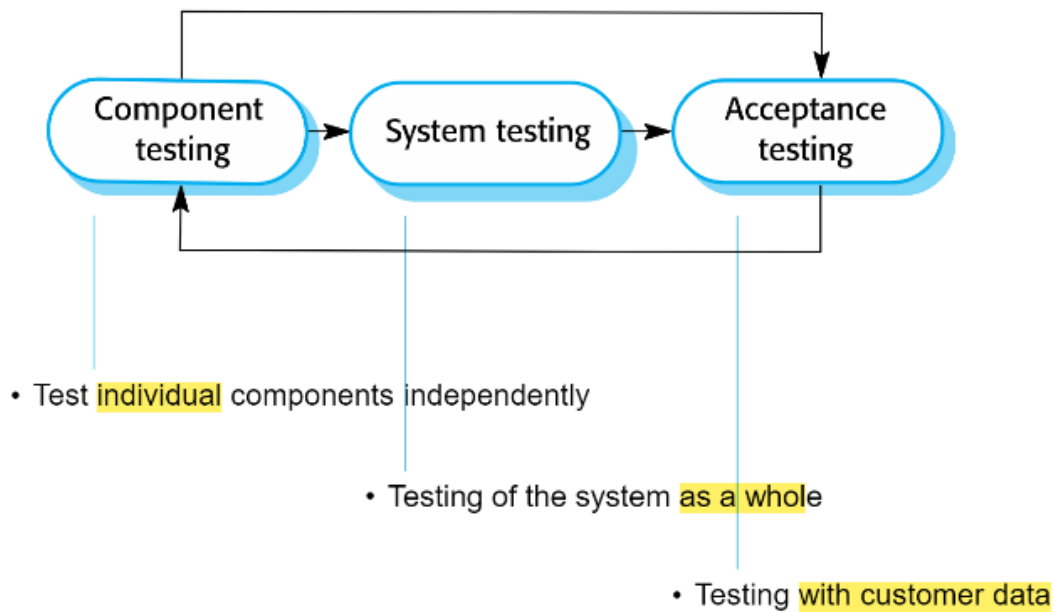
Verification and validation (V & V)

- show that a system conforms to its specification and meets the requirements

Involve checking and review processes and system testing

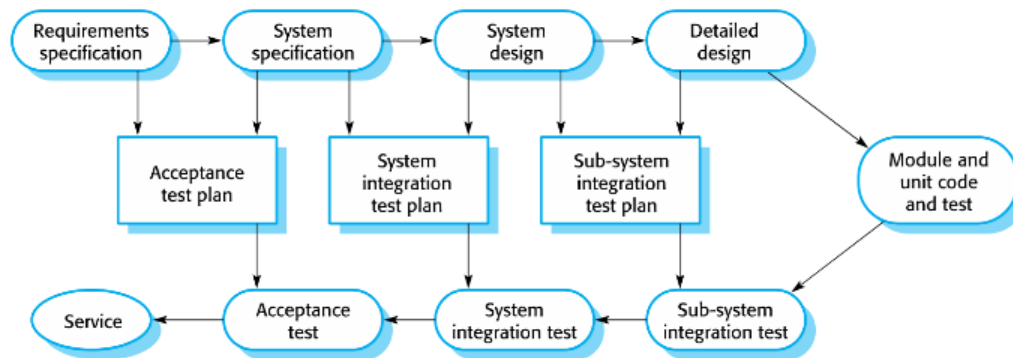
- System testing: executing the system w test cases
- Testing: most commonly V&V

## Stages of testing



## Testing phases in a plan-driven software process

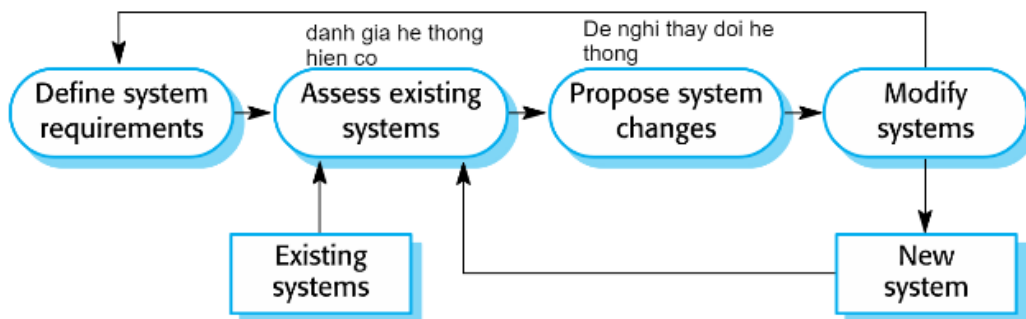




## ACTIVITY: software evolution

Software is inherently flexible and can change

Requirements can change (changing business circumstances) → software must also evolve



## COPING WITH CHANGE

Changes is inevitable in all large software projects

- Business changes
- New technologies
- Changing platforms

Changes leads to rework

- Cost include rework (re-analysing requirements) and implement new functionality

## Software prototyping

A prototype is an initial version of a system and used to demonstrate concepts and try out design options

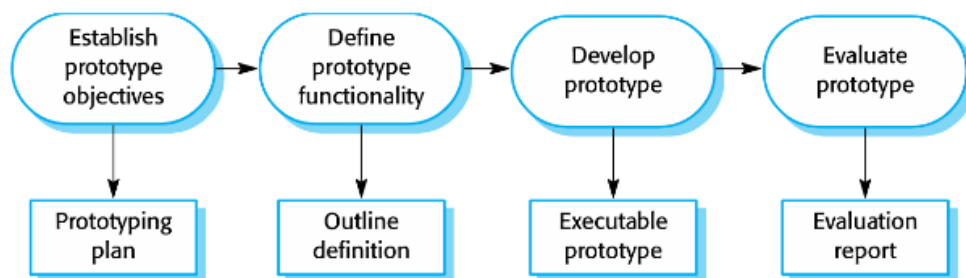
A prototype can be used in:

- requirements engineering process: elicitation and validation
- design process: options and develop UI design
- testing process: run back-to-back tests

### Benefits:

- Improved system usability.
- A closer match to users' real needs.
- Improved design quality.
- Improved maintainability.
- Reduced development effort.

## The process of prototype



### Prototype development:

- May be based on rapid prototyping languages or tools
- May involve leaving out functionality

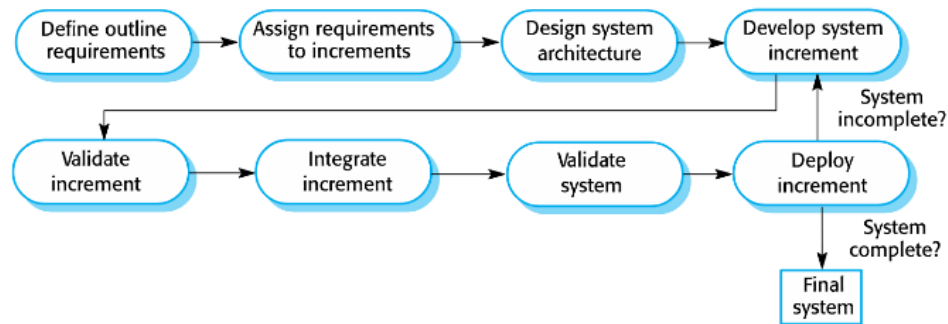
## Incremental delivery

The development and delivery is broken down into increments

- each increment delivering part of the required functionality
- user requirements are prioritised and the highest priority requirements are included in early increments

Two approaches:

- Incremental development: by developer
- Incremental delivery: for end-user



### Advantages:

- system functionality is available earlier.
- early increments act as a prototype
- lower risk of overall project failure.
- highest priority system services receive most testing.

### Problems:

- may require a set of basic facilities cơ sở
- the specification is developed in conjunction with the software. giao thoa

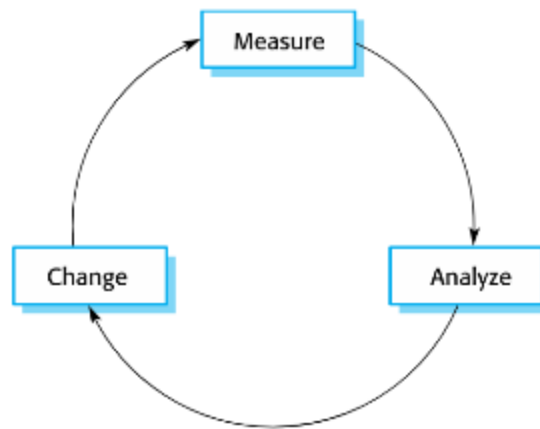
## PROCESS IMPROVEMENT

Software process improvement

- enhancing the quality
- reduce cost
- accelerating development process

Process improvement

- understanding existing processes
- changing these processes



## Process improvement activities

### Process measurement:

Measure 1 or more attributes so you can decide that is the improvements effective?

### Process analysis:

Analyze the process and find the weaknesses or bottleneck. Process model (process map) that describe the process may be developed

### Process change:

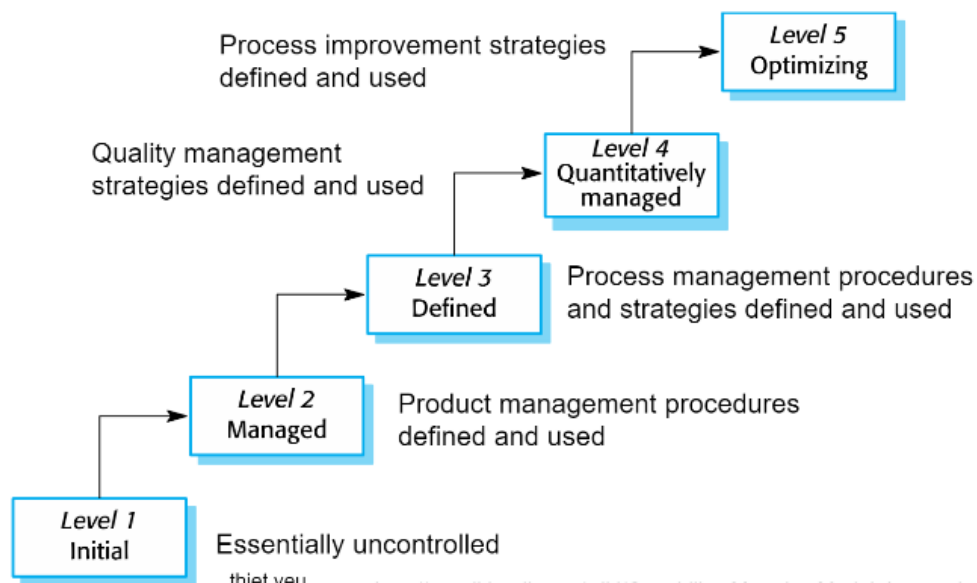
Proposed to address some of the identified process weaknesses. These are introduced and the cycle resumes to collect data about the effectiveness of the changes.

## The capability maturity model (CMM)

Capability Maturity Model Integration (CMMI) is a process level improvement training appraisal program

CMMI defines the most important elements that requires to build great outcome  
Its require by many U.S Governments contracts

## The capability maturity model (CMM)



## Software project documentation

Activity	Document
Validation & Verification	<b>SVVP</b> - Software Validation & Verification Plan
Quality Assurance	<b>SQAP</b> - Software Quality Assurance Plan
Configuration	<b>SCMP</b> - Software Configuration Management Plan
Project status	<b>SPMP</b> - Software Project Management Plan
Requirements	<b>SRS</b> - Software Requirements Specifications
Design	<b>SDD</b> - Software Design Document / Software Detail Design Document
Code	Source <b>Code</b>
Testing	<b>STD</b> - Software Test Document
Operation	User's <b>Manual</b>