

# Software Process Models

1

## What is a Process

*"A process defines who is doing what, when, and how to reach a certain goal."*

Ivar Jacobson, Grady Booch, and James Rumbaugh

2

## What is a Process?

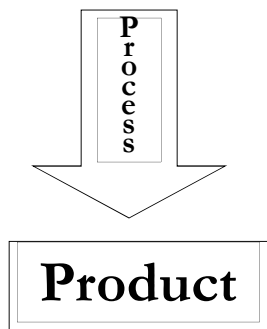
- Given *input*, *transforms* it into *output*
- Consist of a set of activities
- Ordering among the activities (a partial order)
- Software Process
  - Also called *methodology* sometimes
  - *People* are everything
  - Involves use of “resources”, such as ???
  - Software process models ~~ software *lifecycle* models
- Process descriptions are also specifications
  - Should be as complete, consistent and clear

3

## Why Software Process?

Quality of product ← ⇌ Quality of Process

- ❑ Garbage in garbage out,  
so get the right requirements



So, know the input sources, specify process & specify product

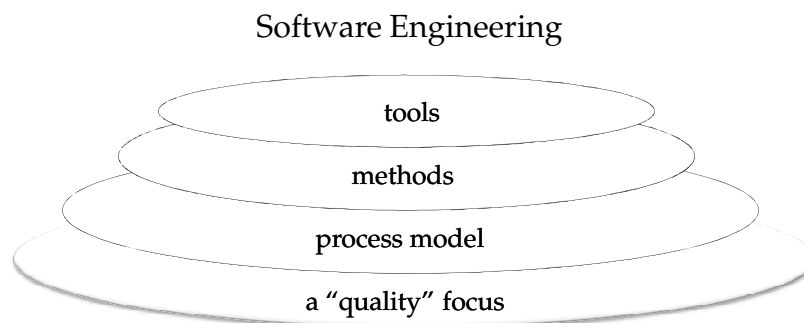
4

## Why Software Process?

- What? A software process - a series of predictable steps that leads to a timely, high-quality product.
- Who? Managers, software engineers, and customers.
- Why? Provides stability, control, and organization to an activity otherwise chaotic activity.
- Steps? A handful of activities are common to all software processes, details vary.
- Work product? Programs, documents, and data.
- Correct process? Assessment, quality deliverable.

5

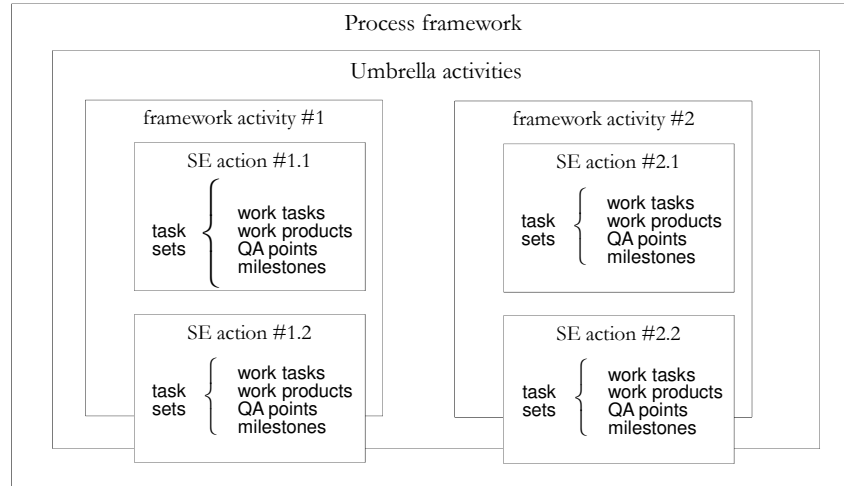
## A Layered Technology



6

# A Process Framework

Software process



7

## Umbrella Activities

- Software project management
- Formal technical reviews
- Software quality assurance
- Software configuration management
- Work product preparation and production
- Reusability management
- Measurement
- Risk management

8

## Framework Activities

- Communication
- Planning
- Modeling
  - Analysis of requirements
  - Design
- Construction
  - Code generation
  - Testing
- Deployment

9

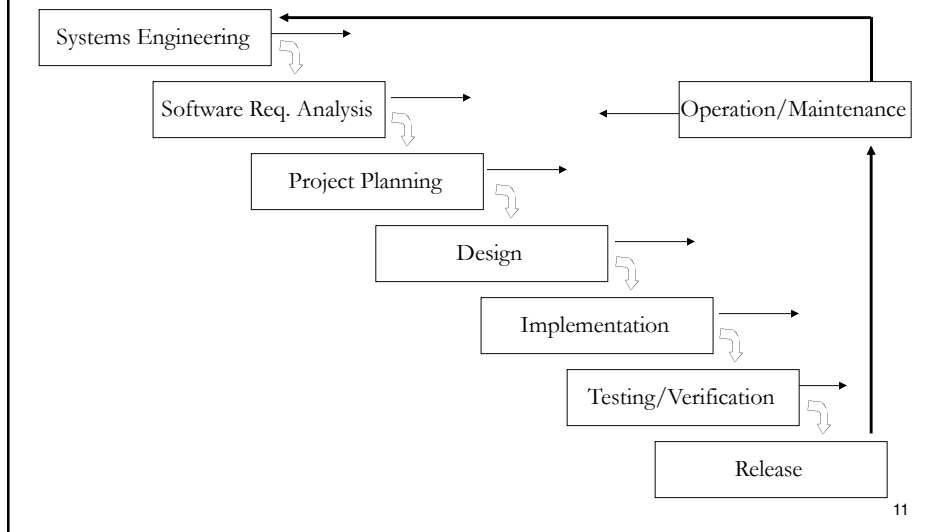
## Software lifecycle models: Generic software process models

- The waterfall model
  - Separate and distinct phases of specification and development
- Incremental development (w. evolutionary)
  - Each increment represents one functionality;
  - Specification and development can be interleaved
- The Spiral model
  - Each loop in the spiral represents an incremental phase in the process.
- Reuse-based development
  - The system is assembled from existing components

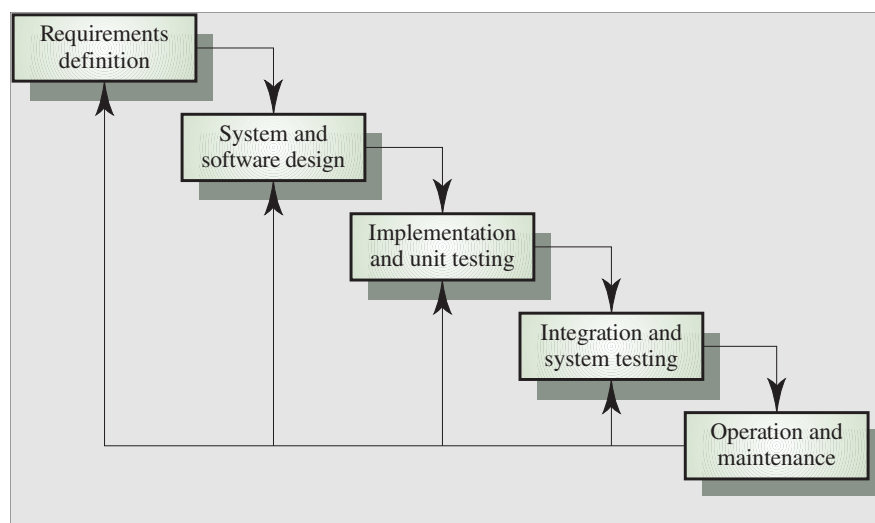
10

## Waterfall model 1 [aka Royce1970]

*Separate and distinct phases of specification and development*



## Waterfall model 2 [Sommerville2000]

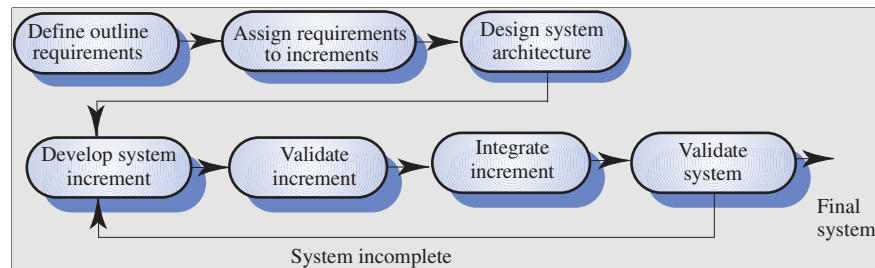


## Waterfall model – pros and cons

- Inflexible partitioning of the project into distinct stages
- This makes it difficult to respond to changing customer requirements
- This model is only appropriate when the requirements are well-understood
- *Big design up front* - time spent early on making sure that requirements and design are correct is very useful in economic terms (it will save you much time and effort later)
- Simplicity and controllability

13

## Incremental development



- Development and delivery is broken down into increments
- Each increment delivers part of the required functionality
- Requirements are prioritised and the highest priority requirements are included in early increments
- Once the development of an increment is started, the requirements are frozen
  - Requirements for later increments can continue to evolve

14

## Incremental development advantages

- System functionality is available earlier and customer does not have to wait as long
- Early increments act as a prototype to help elicit requirements for later increments
- Lower risk of overall project failure
- The highest priority system services tend to receive the most testing

15

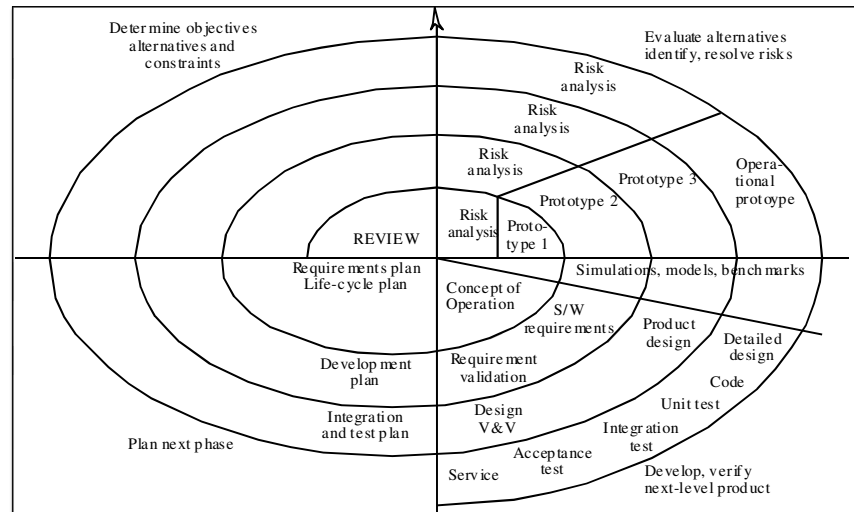
## Spiral development

- Process is represented as a spiral rather than as a sequence of activities with backtracking
- Each ***loop*** in the spiral represents a ***phase*** in the process.
- ***No fixed phases*** such as specification or design - loops in the spiral are chosen depending on what is required
- ***Risks*** are explicitly assessed and resolved throughout the process

16



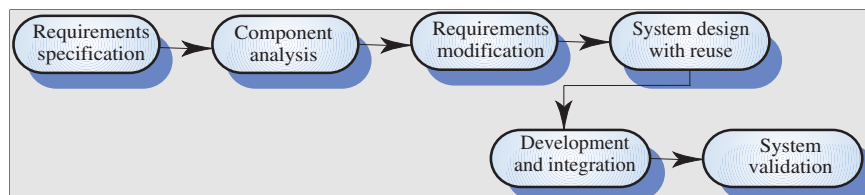
## Spiral Model



17

## Reuse-oriented development

- Based on systematic reuse where systems are integrated from existing components
- Process stages
  - Component analysis
  - Requirements modification
  - System design with reuse
  - Development and integration
- Becoming important but still limited experience with it



18

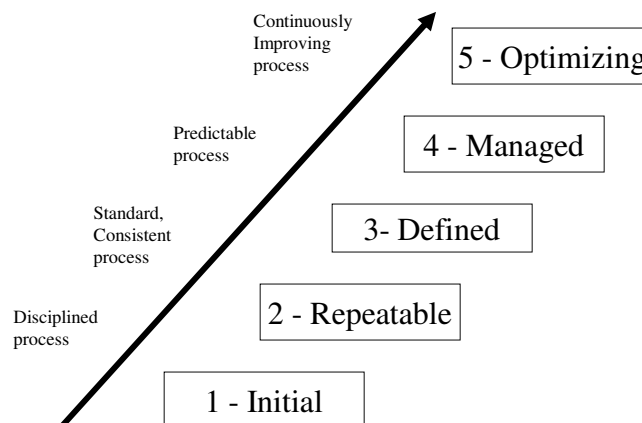
## The CMMI

- The CMMI defines each process area in terms of “specific goals” and the “specific practices” required to achieve these goals.
- Specific goals establish the characteristics that must exist if the activities implied by a process area are to be effective.
- Specific practices refine a goal into a set of process-related activities.

19

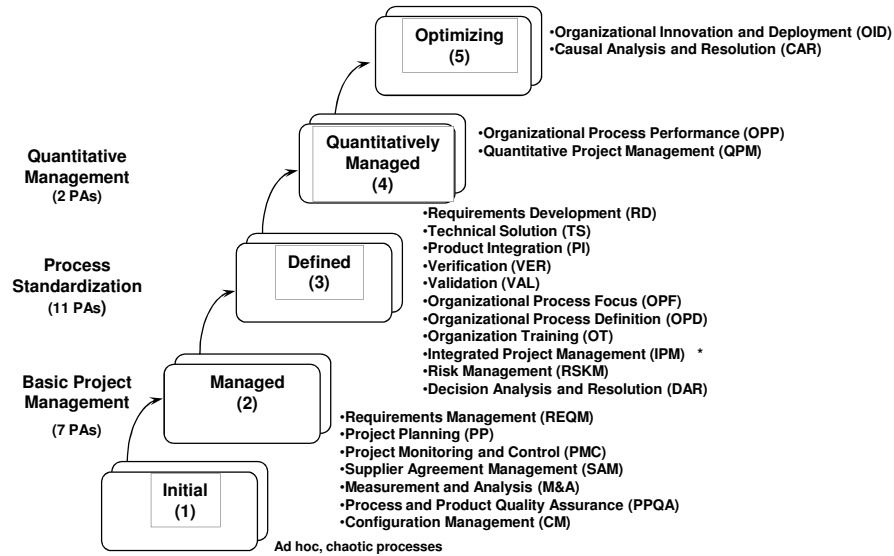
## Capability Maturity Model [SEI]

- Developed in 1986 with leadership from the Software Engineering Institute (SEI) of Carnegie Mellon University (CMU), CMU-SEI.
- For assessing and improving software processes
- As a guidance for measuring software process maturity



20

## CMMI-SE/SW/IPPD/SS - Staged



21

## Some CMM Observations...

- The number of companies using CMM to assess their software management practices more than doubles every 5 years
- Software Quality Assurance is the biggest obstacle for organizations trying to move from level 1 to level 2.
- Organization Process Definition is one of the biggest obstacles for organization trying to move from level 2 to level 3.
- On average, it takes an organization:
  - 25 months to move from level 1 to 2
  - 22 months to move from level 2 to 3
  - 36.5 months to move from level 3 to 4

22

## Some CMM Observations...

- Only 1.2% of companies engaged in CMM have IT departments with over 2000 employees. *Of these large companies, 40% are at CMM levels 3, 4 or 5.*
- About 80% of companies engaged in CMM have IT departments with less than over 300 employees. *Of these smaller companies, 21% are at CMM levels 3, 4, or 5.*
- About 1/3 of companies engaged in CMM are located overseas (primarily India), and are 3 times more likely to reach CMM level 4 or 5 than US organizations.
- *Only about 23% of organizations surveyed eventually move from level 2 to level 3 or higher.*

23

## Software Process Variability

- Software processes may vary radically from one organization to another
- Factors contributing to this variability include
  - Technical maturity
  - Disciplinary involvement
  - Organizational culture
  - Application domain
- There is therefore no 'ideal' software engineering process

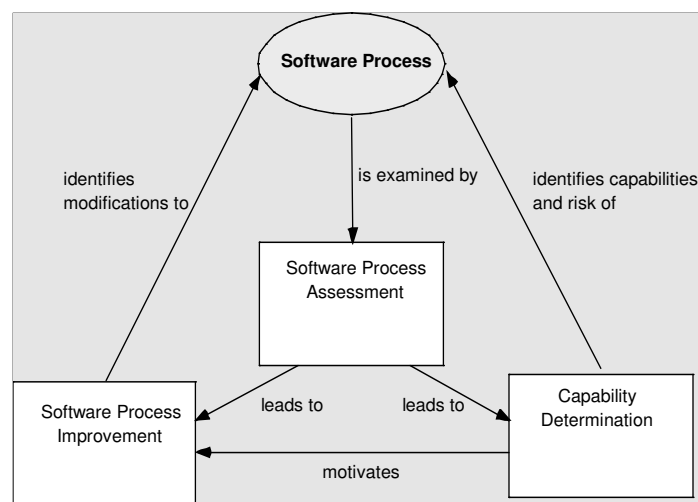
24

## Process Assessment

- The process should be assessed to ensure that it meets a set of basic process criteria that have been shown to be essential for a successful software engineering.
- Many different assessment options are available:
  - SCAMPI
  - CBA IPI
  - SPICE
  - ISO 9001:2000

25

## Assessment and Improvement



26

## The Primary Goal of Any Software Process: *High Quality*

Remember:

High quality  $\Rightarrow$  project timeliness

Why?

Less rework!