CSE4006 Software Engineering

02. Process : A Generic View

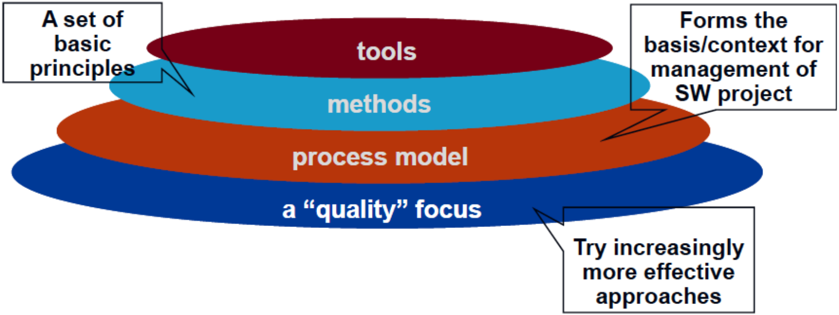
Scott Uk-Jin Lee

Department of Computer Science and Engineering
Hanyang University ERICA Campus

1st Semester 2015

A Layered Technology

action, activity, process, task


A set of basic principles
tools
methods
process model
a “quality” focus
Forms the basis/context for management of SW project
Try increasingly more effective approaches

Software Development Process

- Process
  - Sequence of tasks carried out for certain purpose (IEEE)
- Software Development Process
  - Set of methods, practices and activities used for developing and maintaining a software or related products (CMM)
  - Establishing and operating a process that enables developing a high quality and reliable software is the key component for the competitiveness of a software company
- **The right process will produce the right result**

The Importance of Process

- Every organization tries to “get the fat” out of industrial processes for more than a century

  e.g. Toyota’s cost reduction for vehicle manufacturing

- Process help us order our thinking by defining common activities and artifacts
  - Process is a means to capture and transfer the knowledge we gain in developing a particular product
  - Process improvement identify and deploy knowledge over large groups

The Necessity of Process Improvement

- A process is about incorporating discipline into routine activities to check everything that was supposed to be done was done
  - Make sure
    - There was sufficient repeatability in the tasks to make future work predictable
    - This process repeatability and predictability are called "capability maturity"
- Informally speaking, process improvement is to incorporate individual wisdom/guidance into the way the organization works

The Necessity of Process Improvement

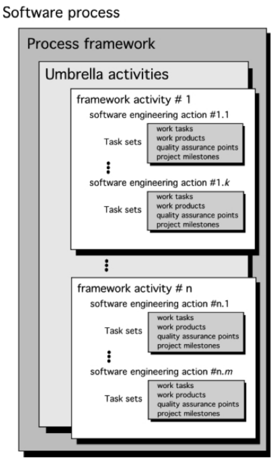
sw가 무형이고 아직 오래 되지 않아서 프로세스가 미성숙

- Quality of software products depends largely on the process used for the development and maintenance (Humphrey, 1995)
- 90% of the problems found in a product are resulted from the the problem within the process (US DoD report 1987)
- Problems of software development: low productivity / quality
  - Delayed delivery and cost overrun
  - US DoD report 2002 (80% of F-22 & 65% of B-2 = software):
    - 16% quality / delivery / cost = satisfactory
    - 53% delayed delivery & cost overrun
    - 31% failure
- a good quality software without a good process?
  - cannot plan or manage quality
  - cannot good quality products repeatedly

Software Process Improvement in Korea

- National IT Industry Promotion Agency (www.nipa.kr): software process quality certification
  - Assess and certify software development process capability level by analyzing the capability of enterprise (organization) in terms of step-by-step tasks and deliverables managements in software development
  - Process level of domestic software companies in 2007 = 1.6655 (CMMI of 91 companies)
  - Percentage of medium sized companies in domestic software industry (4937/4986, '06KAIT)
  - International standards are only suitable for large companies: complex procedures, high costs, discrepancy in domestic IT industry and medium sized software companies → decreased effectiveness of certification

Generic Process Model & Process Flow



Software process

Process framework

Umbrella activities

framework activity # 1

software engineering action #1.1

Task sets

work tasks
work products
quality assurance points
project milestones

⋮

software engineering action #1.*k*

Task sets

work tasks
work products
quality assurance points
project milestones

⋮

framework activity # n

software engineering action #n.1

Task sets

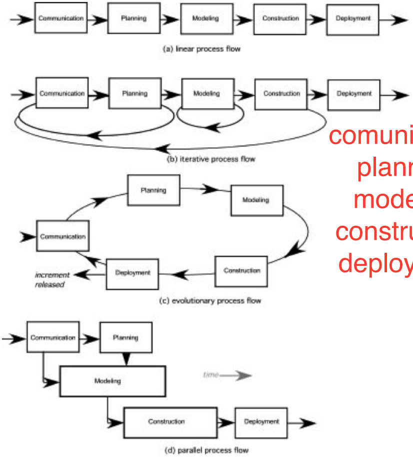
work tasks
work products
quality assurance points
project milestones

⋮

software engineering action #n.*m*

Task sets

work tasks
work products
quality assurance points
project milestones



(a) linear process flow

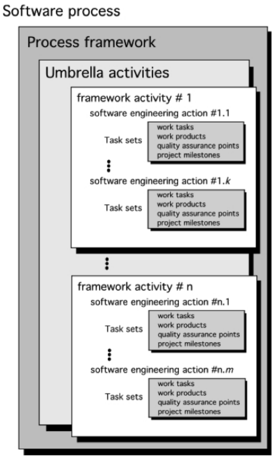
(b) iterative process flow

(c) evolutionary process flow

(d) parallel process flow

comunication
planning
modeling
construction
deployment

Process Framework



Software process

Process framework

Umbrella activities

framework activity # 1

software engineering action #1.1

Task sets

work tasks
work products
quality assurance points
project milestones

⋮

software engineering action #1.*k*

Task sets

work tasks
work products
quality assurance points
project milestones

⋮

framework activity # n

software engineering action #n.1

Task sets

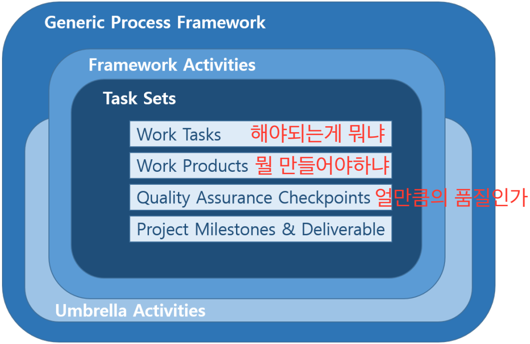
work tasks
work products
quality assurance points
project milestones

⋮

software engineering action #n.*m*

Task sets

work tasks
work products
quality assurance points
project milestones



Generic Process Framework

Framework Activities

Task Sets

Work Tasks 해야되는게 뭐냐

Work Products 뭘 만들어야하냐

Quality Assurance Checkpoints 얼만큼의 품질인가

Project Milestones & Deliverable

Umbrella Activities

Framwork Activities

- Communication
  - **Elicitation of requirements**
    - Work tasks
    - Work products
    - QA checkpoints
    - Project milestones & deliverables
  - ...
- Planning
  - ...
- Modeling
  - Analysis of requirements
  - Design
- Construction
  - Code Generation
  - Testing
- Deployment

work tasks

1. reading assignment sheet
2. send email to prof for assuring spec
3. visit prof

lab(se);

Umbrella Activities

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

The Process Model - Adaptability

- the framework activities will **always** be applied on **every** project ... BUT
- the tasks (and degree of rigor) for each activity will vary based on:
  - the type of project
  - characteristics of the project
  - common sense judgement; concurrence of the project team

The Process Model - Adaptability

stakeholder list up
meeting scheduling(큰 프로젝트의 경우 따로따로 만남)
function feature list up

- the framework activities will **always** be applied on **every** project ... BUT
- the tasks (and degree of rigor) for each activity will vary based on:
    - the type of project
    - characteristics of the project
    - common sense judgement; concurrence of the project team

Software Capability Management Model (CMM)

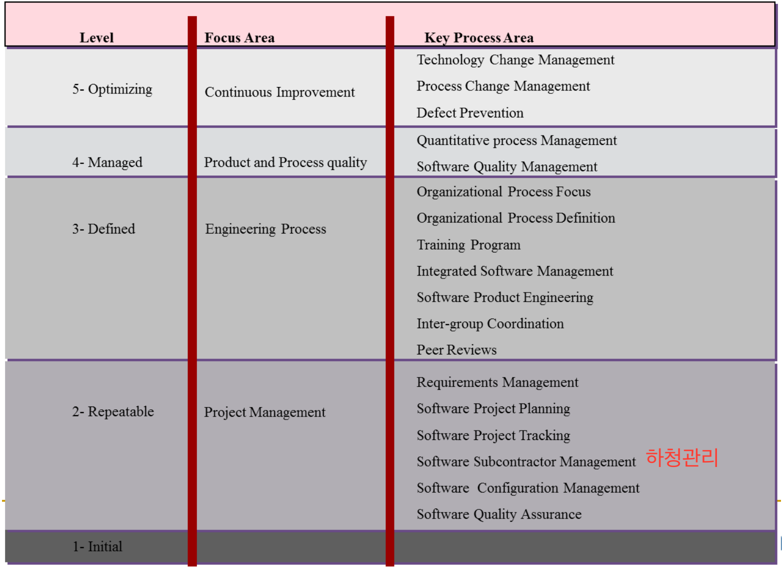
- SEI @ Carnegie Mellon University (CMU) 1991
- Quantifies ability to consistently & predictably develop high quality software
- Includes procedures for screening and assessment
- CMM level of 3 is required for bidding for software development of U.S. government agencies
  - widely used despite the controversies (for and against CMM)
  - utilized as bidder qualification and evaluation
  - typically utilized in the North American industries

The Capability Maturity Model Integrated (CMMI)

CMM upgrade ver

- capability maturity = the process repeatability and predictability
- 2^{nd} generation of CMMs
- developed by U.S. DoD and SEI @ CMU as a common and extensible framework
  - By mid-90's, five-level world view of CMM for Software became dominant and there appeared to many CMMs
  - Integrated model to go against ISO/IEC adopting European SPICE model as the international standard (ISO/IEC15504) of process model
- CMMI solutions
  - CMMI for Development (CMMI-DEV)
    - product and service development
  - CMMI for Services (CMMI-SVC)
    - service establishment, management, and delivery
  - CMMI for Acquisition (CMMI-ACQ)
    - product and service acquisition

Key Process Area (KPA) for each Level of CMMI



Level	Focus Area	Key Process Area
5- Optimizing	Continuous Improvement	Technology Change Management
		Process Change Management
		Defect Prevention
4- Managed	Product and Process quality	Quantitative process Management
		Software Quality Management
3- Defined	Engineering Process	Organizational Process Focus
		Organizational Process Definition
		Training Program
		Integrated Software Management
		Software Product Engineering
		Inter-group Coordination
		Peer Reviews
2- Repeatable	Project Management	Requirements Management
		Software Project Planning
		Software Project Tracking
		Software Subcontractor Management 하청관리
		Software Configuration Management
		Software Quality Assurance
1- Initial		

KPA of CMMI

- KPA for each level of CMMI
  - CMMI defined **specific goals** & **general goals** for each KPA
  - CMMI defined **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

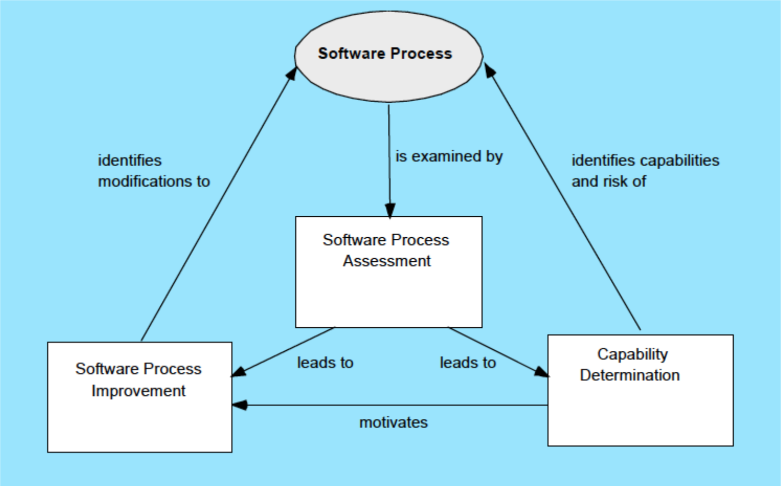
- lower level activities are satisfied at the higher level

lab(se);

Process Assessment

- 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
- **Standard CMMI Appraisal Method for Process Improvement (SCAMPI)**
  - evaluates detailed process area based on CMMI
  - provides benchmark quality rating
- **CMM-Based Appraisal for Internal Process Improvement (CBAIPI)**
  - screening method of SEI on CMM
- **SPICE (ISO/IEC15504)**
  - international standards for software process assessment
- **ISO 9001:2000**
  - international standards for quality management systems

Assessment and Improvement


Software Process
is examined by
identifies modifications to
identifies capabilities and risk of
Software Process Assessment
leads to
leads to
Software Process Improvement
Capability Determination
motivates

Personal & Team Software Process

Personal Software Process

- Recommends five framework activities:
  - Planning
  - High-level design
  - High-level design review
  - Development
  - Postmortem
- stresses the need for each software engineer to identify errors early and to understand the types of errors

Team Software Process

- Each project is **launched** using a **script** that defines the tasks to be accomplished
- Teams are self-directed
- Measurements is encouraged
- Measures are analyzed with the intent of improving the team process

Similar International Standards

Evaluation Assurance Level (EAL)

- Common Criteria (CC) : framework for evaluating and certifying security of an IT product or system
  - recognize as one of the main quality standards for IT security products by governments and IT professionals worldwide
  - enacted as evaluation standard in 1996 to integrate different assessment criteria of different countries and mutually authenticate the evaluation results
  - approved to be an international standard (ISO/IEC 15408) in June 1999
- Korea registered to Common Criteria Recognition Agreement (CCRA) in 2006

Similar International Standards

Evaluation Assurance Level (EAL)

- assignment of numerical grade to an IT product or system following the completion of a **Common Criteria (CC)** security evaluation
- defines 7 levels based on the security requirements defined in CC
- provides different level of confidence depending on whether the system's **principal security features** are reliably implemented
- does **NOT** measure the security of the system itself, BUT simply states at what level the system was **tested** to see if it meets all the requirements of its protection profile
- to achieve a particular EAL, the computer system must meet specific assurance requirements, involving **design documentation, design analysis, functional testing, or penetration testing**

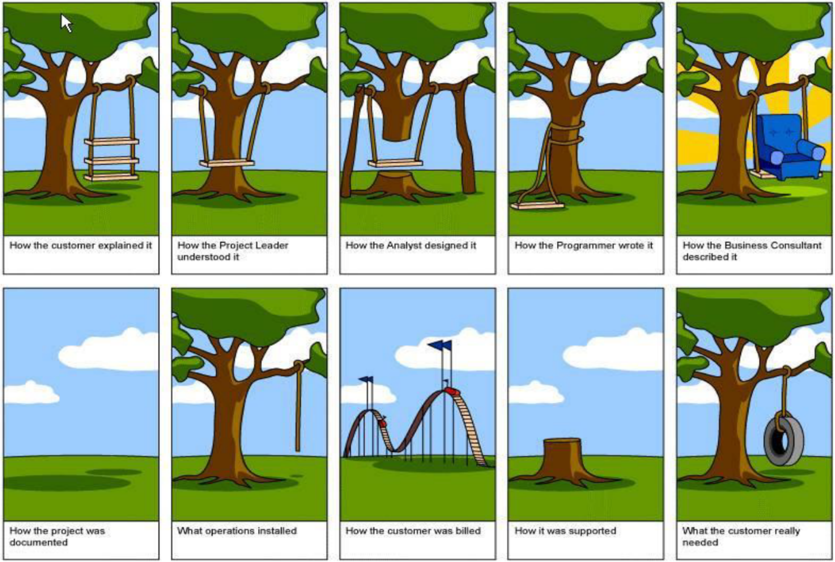
lab(se);

EAL 7 Levels

- **EAL1**: Functionally Tested
- **EAL2**: Structurally Tested
- **EAL3**: Methodically Tested and Checked
- **EAL4**: Methodically Designed, Tested, and Reviewed
  - Commercial operating systems that provide conventional, user-based security features are typically evaluated at EAL4
    - AIX, HP-UX, FreeBSD, Solaris, Novell NetWare, SUSE Linux Enterprise Server 9, SUSE Linux Enterprise Server 10, Windows 2000 Service Pack 3, and Red Hat Enterprise Linux 5
    - Ahnlab 수호신 Absolute 2009 acquired EAL4

EAL 7 Levels

- **EAL5** : **Semi-formally** Designed and Tested
  - Numerous smart card devices have been evaluated at EAL5
  - XTS-400 (STOP 6) is a general-purpose operating system at EAL5 augmented
  - LPAR on IBM System z is EAL5 Certified
- **EAL6** : **Semi-formally Verified** Design and Tested
  - Green Hills Software INTEGRITY-178B OS
- **EAL7** : **Formally Verified** Design and Tested
  - Tenix Interactive Link Data Diode Device


How the customer explained it
How the Project Leader understood it
How the Analyst designed it
How the Programmer wrote it
How the Business Consultant described it
How the project was documented
What operations installed
How the customer was billed
How it was supported
What the customer really needed