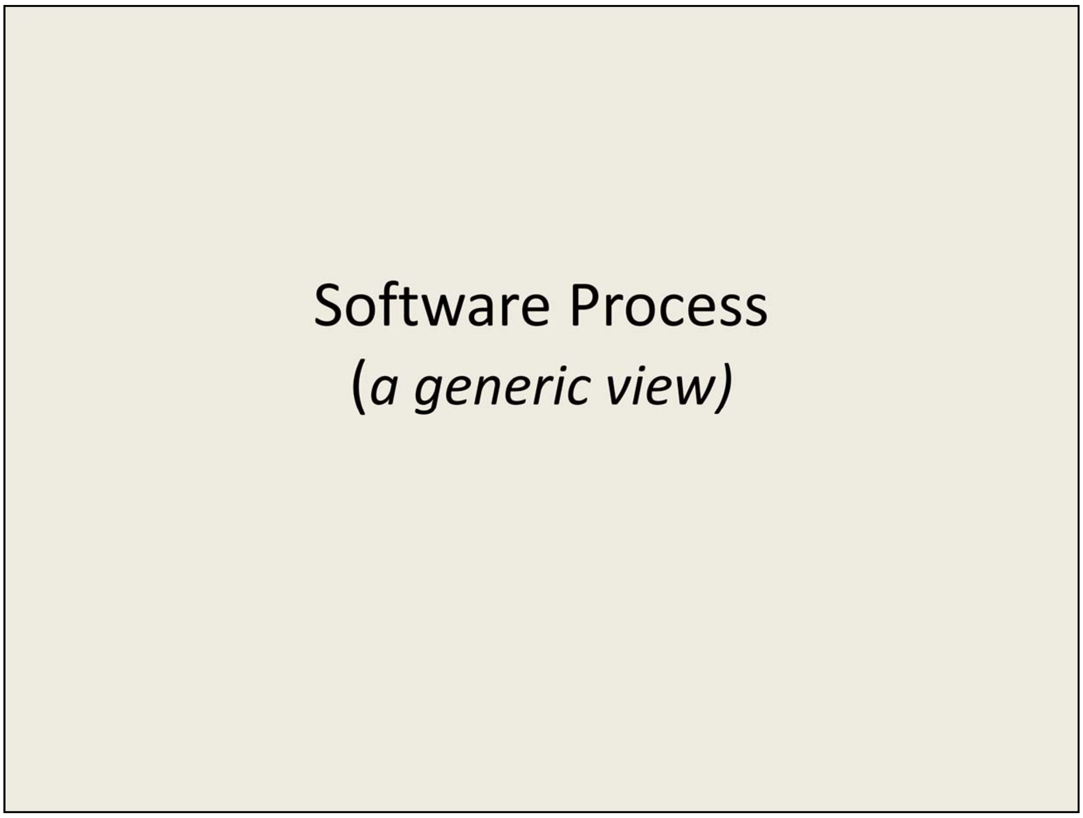
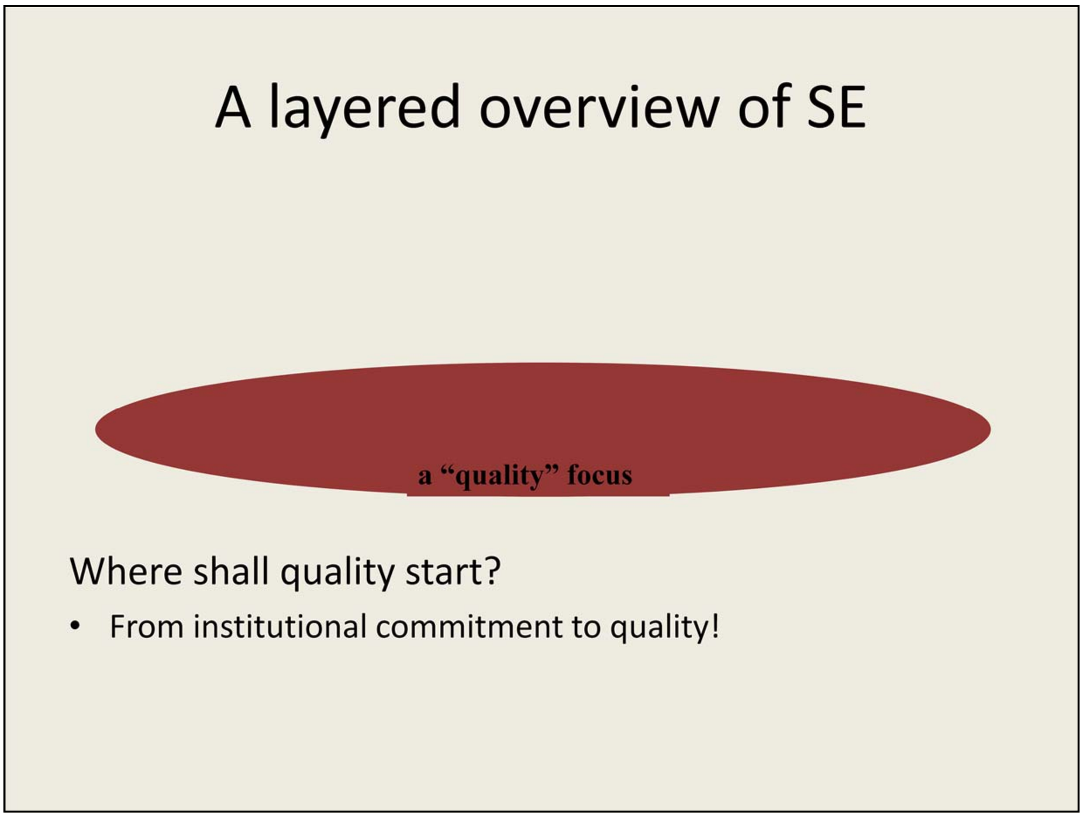


Software Process
(*a generic view)*

A layered overview of SE

Software engineering is to provide processes, methods, tools for improving productivity of software development and quality of finish products.

- Focus on quality!
  - Quality has to be built into every phase of software development.
    - It cannot be an after-thought.
    - Quality means that we should have better quality of software artifacts as the output of each phase of development.
  - And more importantly, we shall have a software development process and quality mandate built into a development organization.



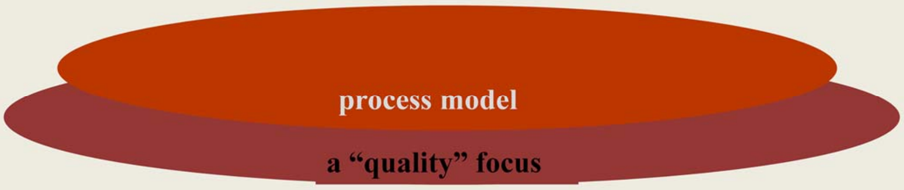
A layered overview of SE

a “quality” focus

Where shall quality start?

- From institutional commitment to quality!

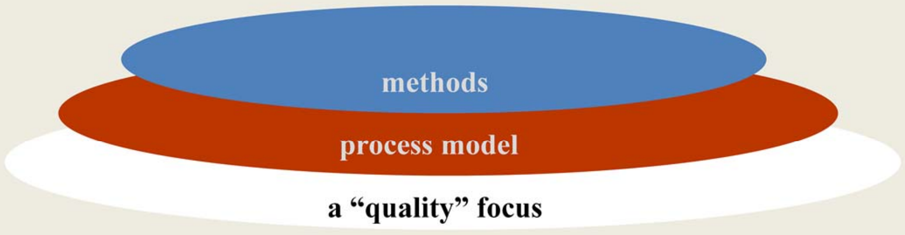
A layered overview of SE



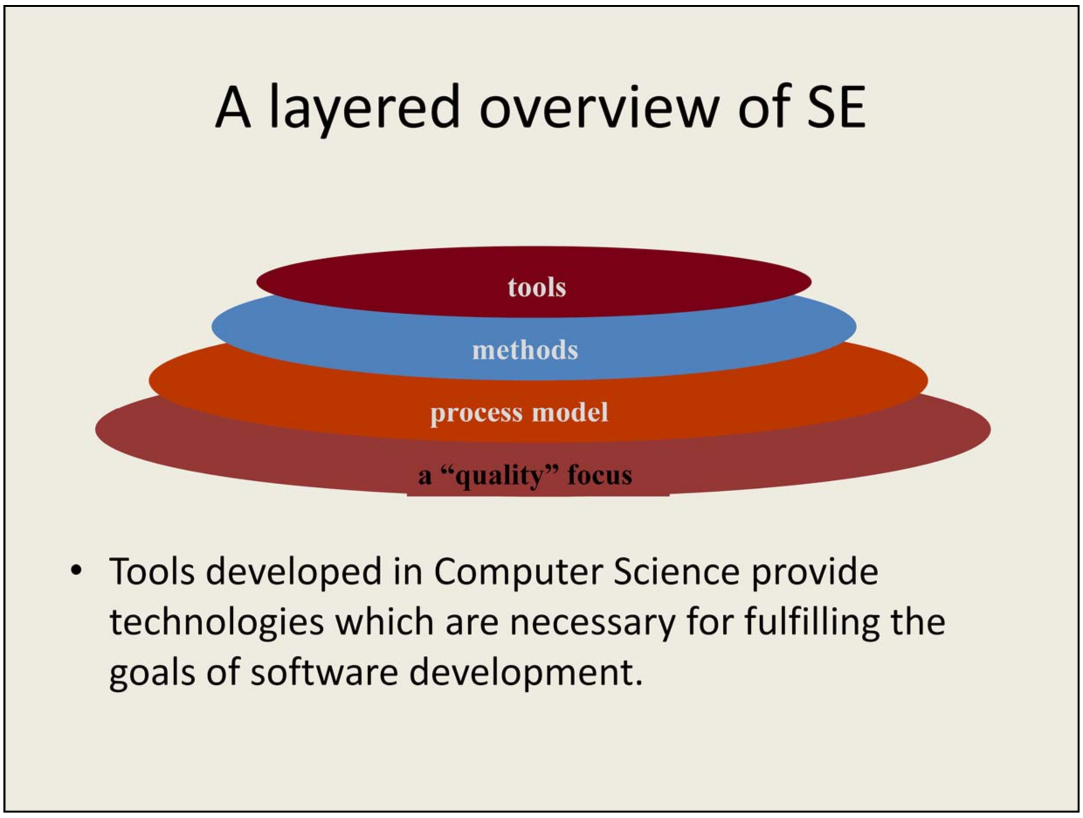
Where shall software engineering start?

- A software development process takes quality as one of its mandates.
- Software process will also serve as a framework which integrates methods and technologies to fulfill its mandate.

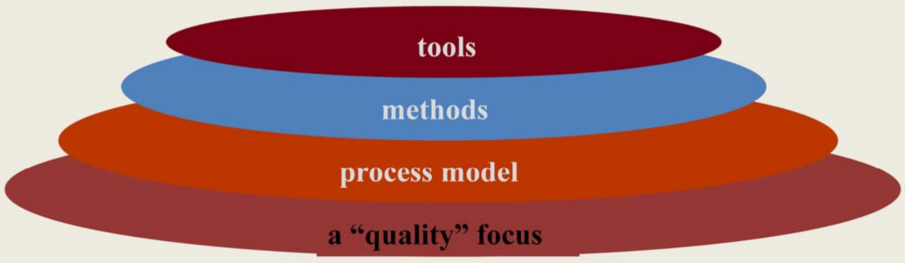
A layered overview of SE



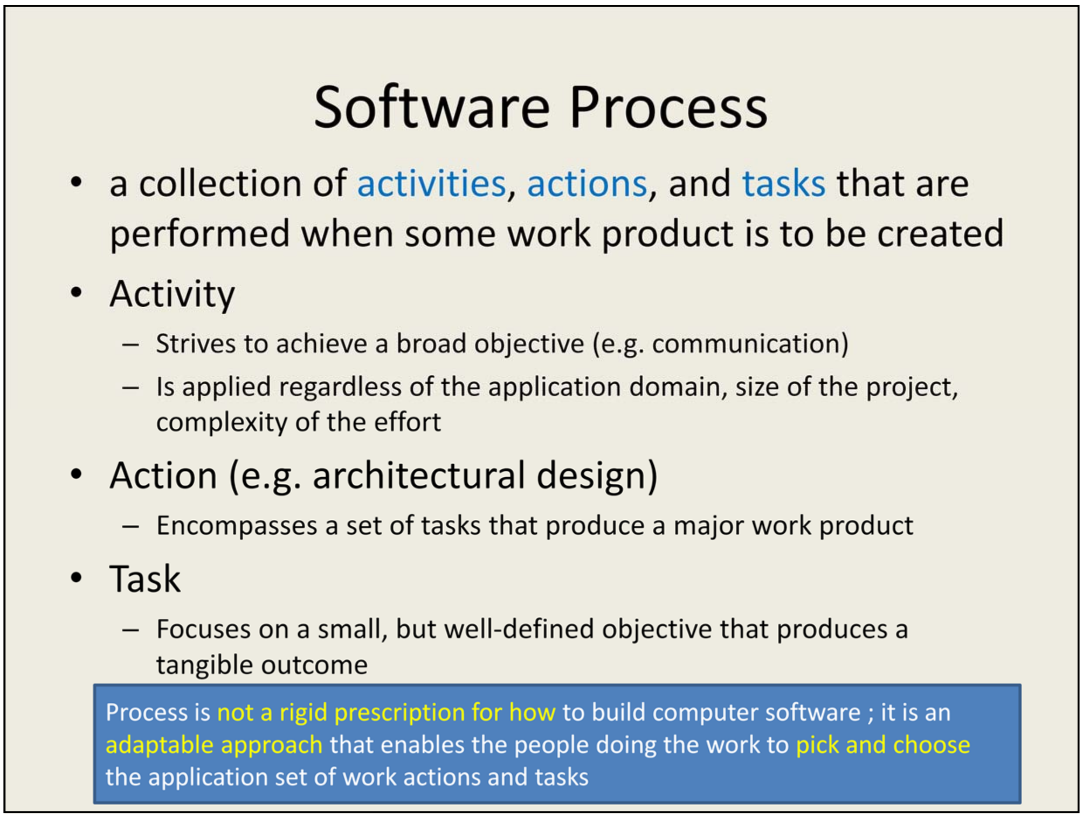
- Methods supply best practices in software engineering for implementing process models and streamlining software development.



A layered overview of SE



- Tools developed in Computer Science provide technologies which are necessary for fulfilling the goals of software development.



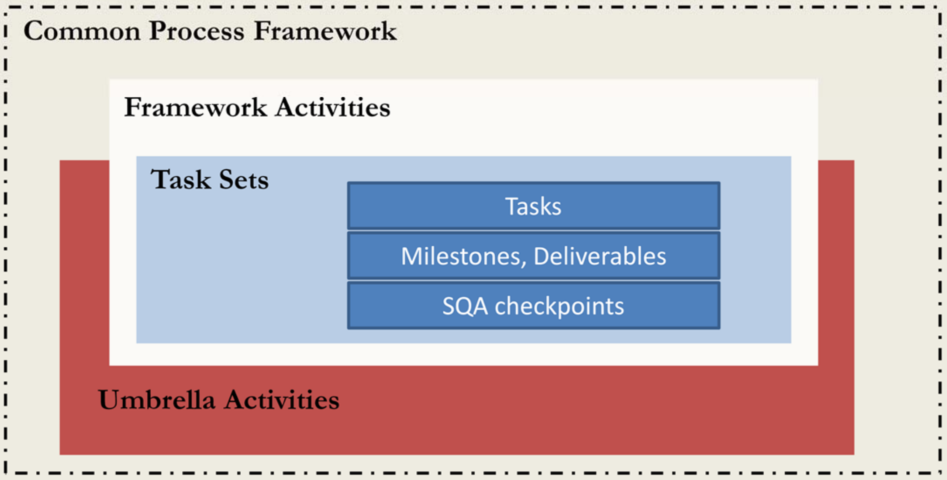
Software Process

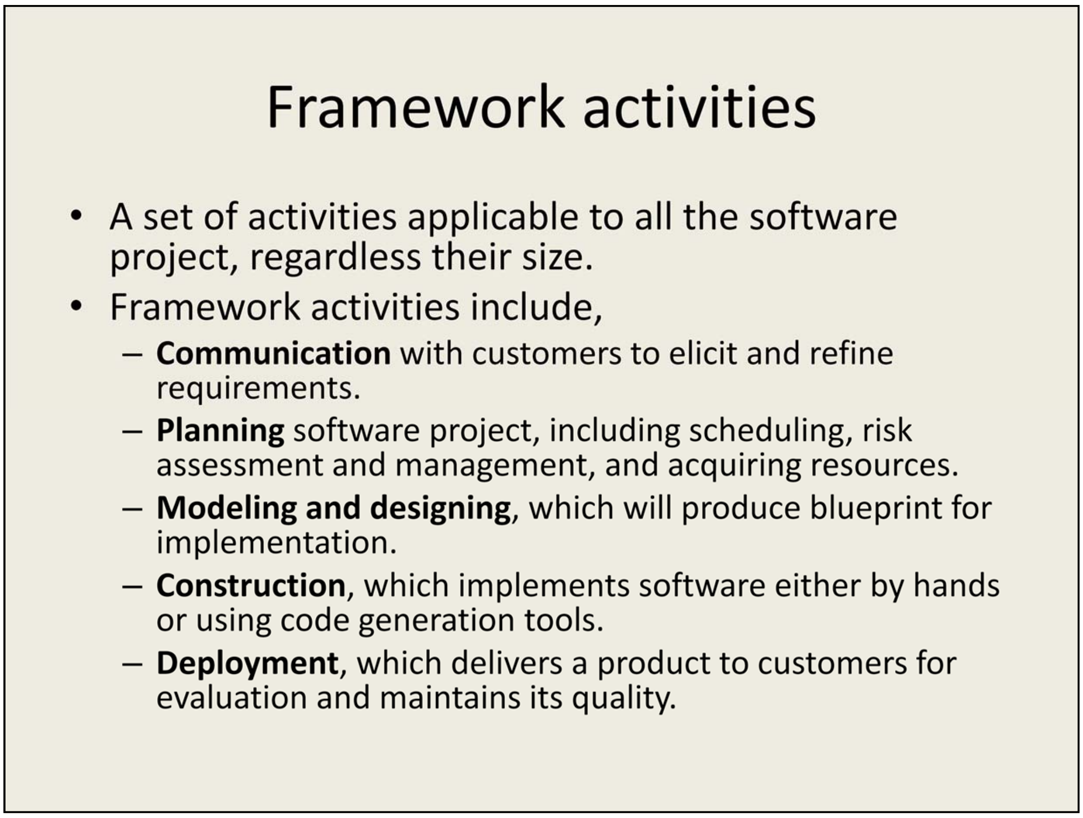
- a collection of activities, actions, and tasks that are performed when some work product is to be created
- Activity
  - Strives to achieve a broad objective (e.g. communication)
  - Is applied regardless of the application domain, size of the project, complexity of the effort
- Action (e.g. architectural design)
  - Encompasses a set of tasks that produce a major work product
- Task
  - Focuses on a small, but well-defined objective that produces a tangible outcome

Process is not a rigid prescription for how to build computer software ; it is an adaptable approach that enables the people doing the work to pick and choose the application set of work actions and tasks

Process Framework

- Process framework establishes the foundation for a complete software process.
  - It identifies both framework activities and umbrella activities.



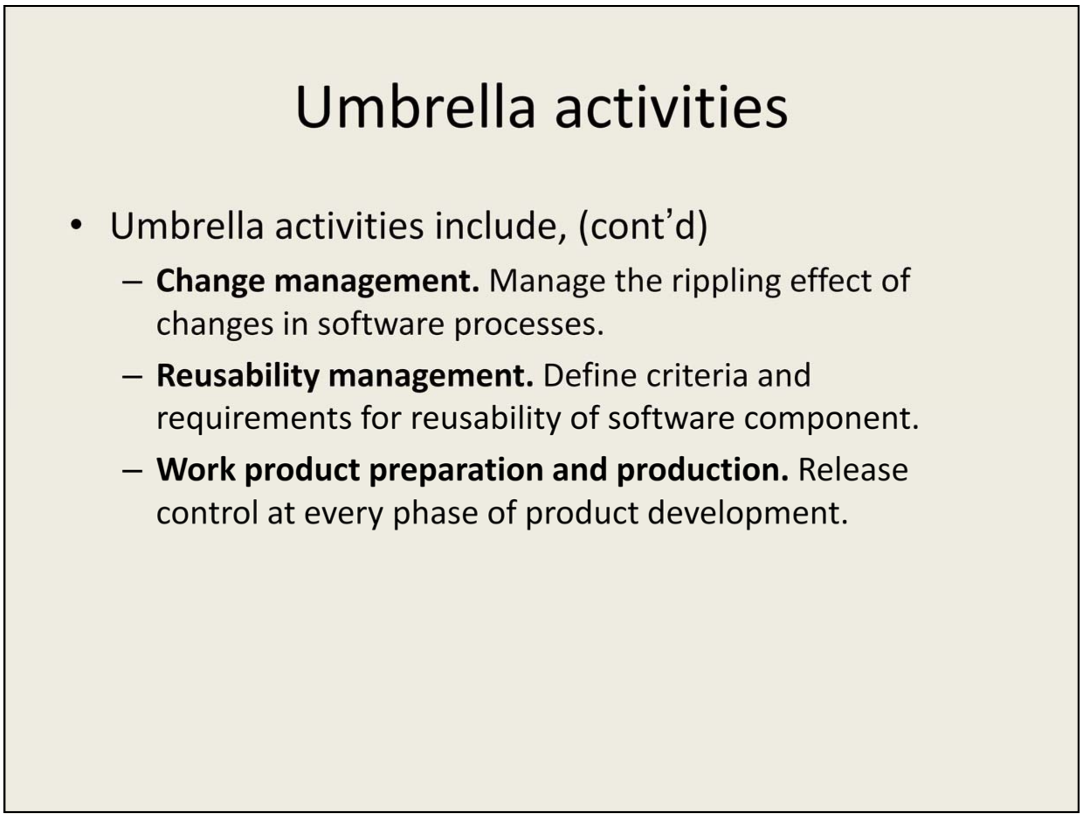


Framework activities

- A set of activities applicable to all the software project, regardless their size.
- Framework activities include,
  - **Communication** with customers to elicit and refine requirements.
  - **Planning** software project, including scheduling, risk assessment and management, and acquiring resources.
  - **Modeling and designing**, which will produce blueprint for implementation.
  - **Construction**, which implements software either by hands or using code generation tools.
  - **Deployment**, which delivers a product to customers for evaluation and maintains its quality.

Umbrella activities

- A set of activities occurring in all the phases of software development.
- Umbrella activities include,
  - **Software project tracking and control.** Check the progress of a project and maintain its schedule.
  - **Risk management.** Assess risks which may affect product quality and schedule.
  - **Software quality assurance.** Quality assurance shall be built into every phase of development.
  - **Formal technical review**, e.g., design review, code review, etc.
  - **Measurement**, quantity software development process such as quality metrics etc.



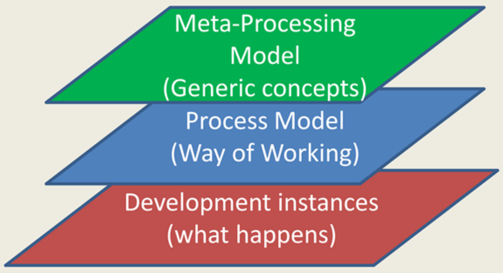
Umbrella activities

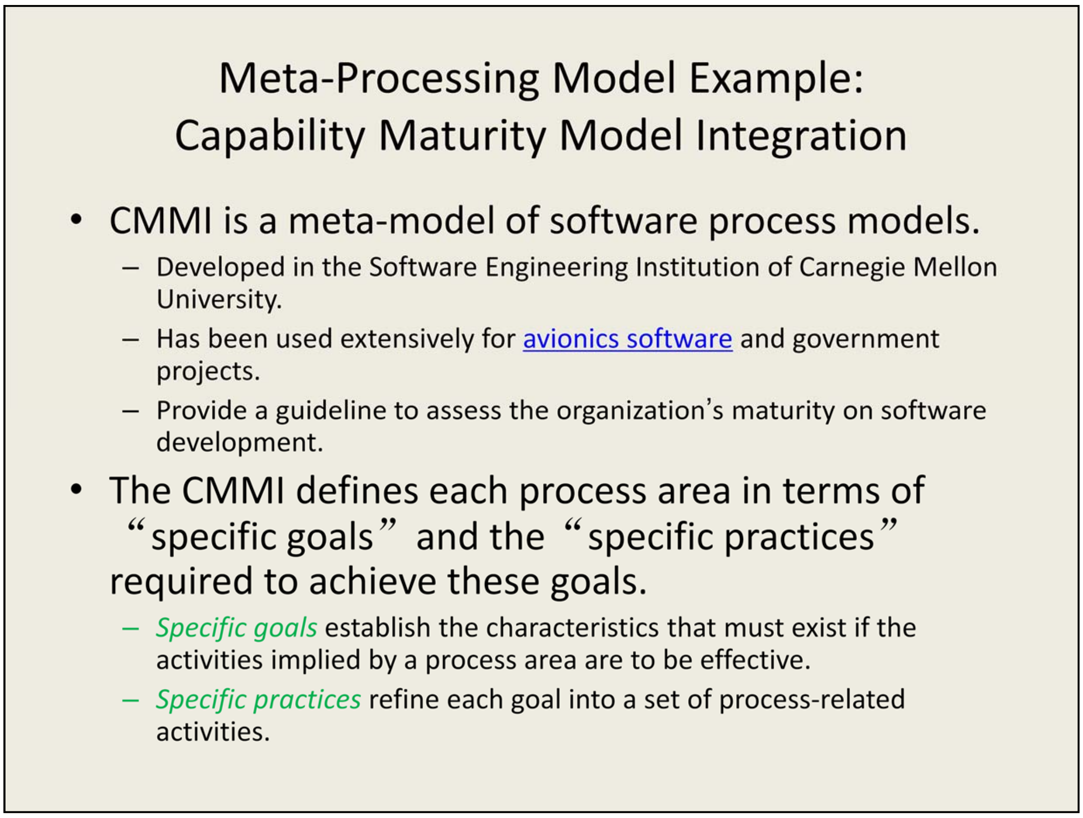
- Umbrella activities include, (cont’d)
  - **Change management.** Manage the rippling effect of changes in software processes.
  - **Reusability management.** Define criteria and requirements for reusability of software component.
  - **Work product preparation and production.** Release control at every phase of product development.

Meta-Model for Software Process

A meta-model is a model of all process models.

- Not bounded to a particular process model;
  - An abstract of elements shared by all the process models;
  - Concepts such as process framework generalizes a set of activities which are required for successful execution of development project.


Meta-Processing Model (Generic concepts)
Process Model (Way of Working)
Development instances (what happens)

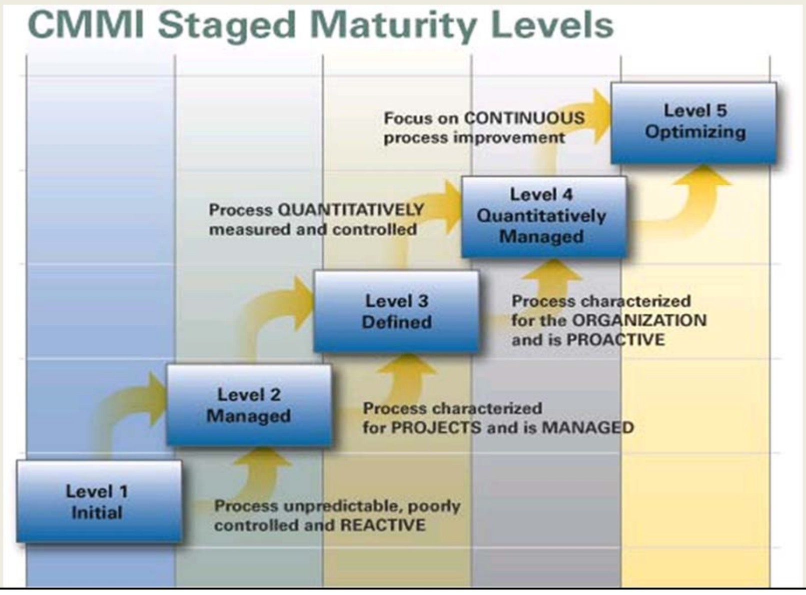


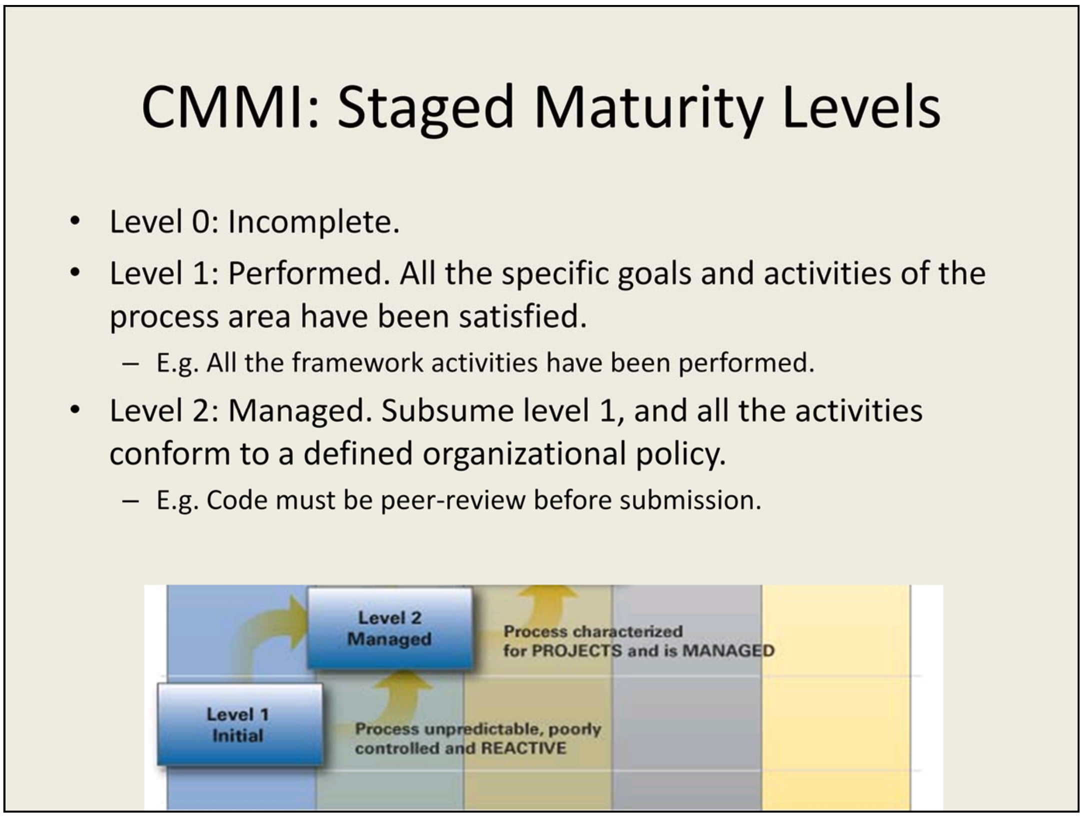
Meta-Processing Model Example: Capability Maturity Model Integration

- CMMI is a meta-model of software process models.
  - Developed in the Software Engineering Institution of Carnegie Mellon University.
  - Has been used extensively for avionics software and government projects.
  - Provide a guideline to assess the organization's maturity on software development.
- 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 each goal into a set of process-related activities.

CMMI: Staged Maturity Levels

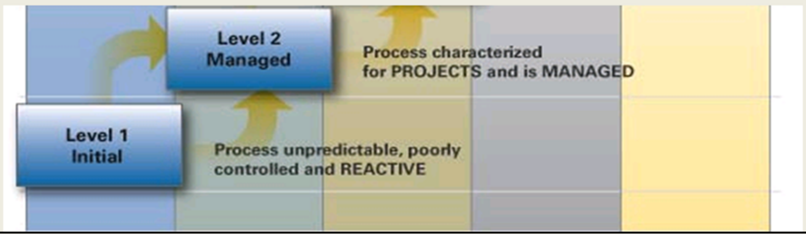
CMMI quantifies maturity of software development process into 6 levels.


CMMI Staged Maturity Levels
Focus on CONTINUOUS process improvement
Level 5 Optimizing
Process QUANTITATIVELY measured and controlled
Level 4 Quantitatively Managed
Level 3 Defined
Process characterized for the ORGANIZATION and is PROACTIVE
Level 2 Managed
Process characterized for PROJECTS and is MANAGED
Level 1 Initial
Process unpredictable, poorly controlled and REACTIVE



CMMI: Staged Maturity Levels

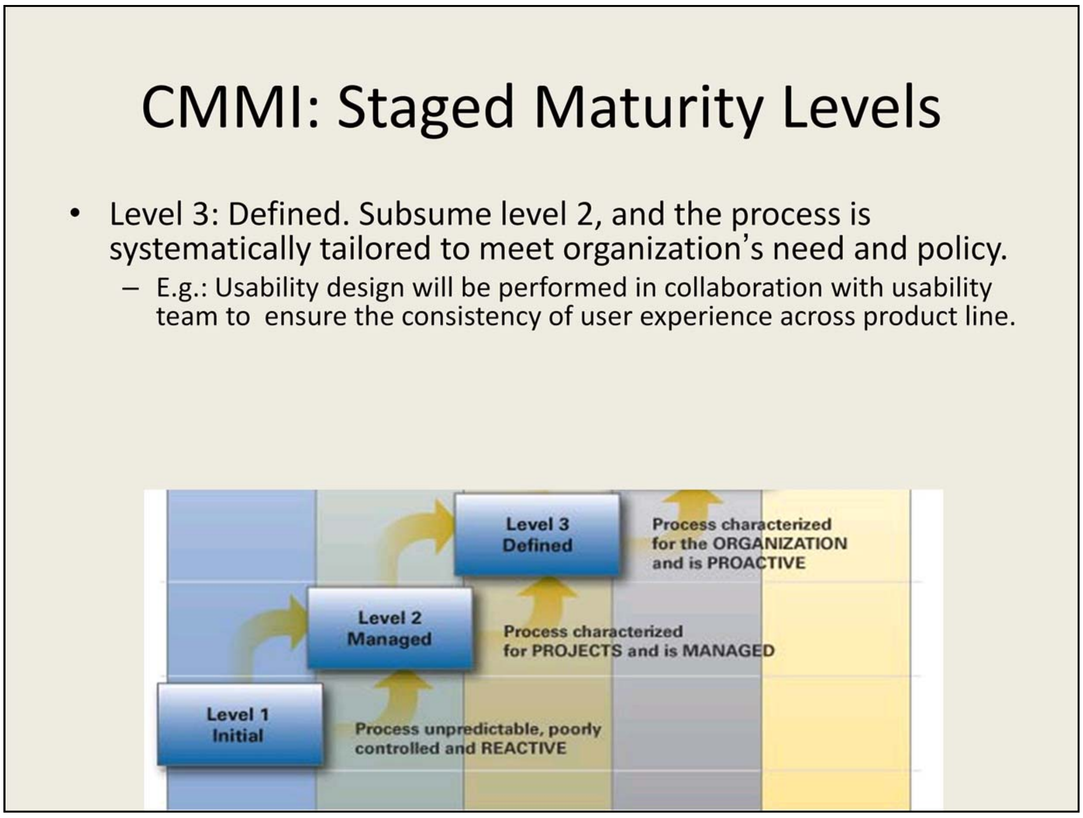
- Level 0: Incomplete.
- Level 1: Performed. All the specific goals and activities of the process area have been satisfied.
  - E.g. All the framework activities have been performed.
- Level 2: Managed. Subsume level 1, and all the activities conform to a defined organizational policy.
  - E.g. Code must be peer-review before submission.


Level 2
Managed
Process characterized for PROJECTS and is MANAGED
Level 1
Initial
Process unpredictable, poorly controlled and REACTIVE

0: The process area, e.g. requirements management is either not performed or does not achieve all goals and objectives defined by the CMMI for level 1 capability.

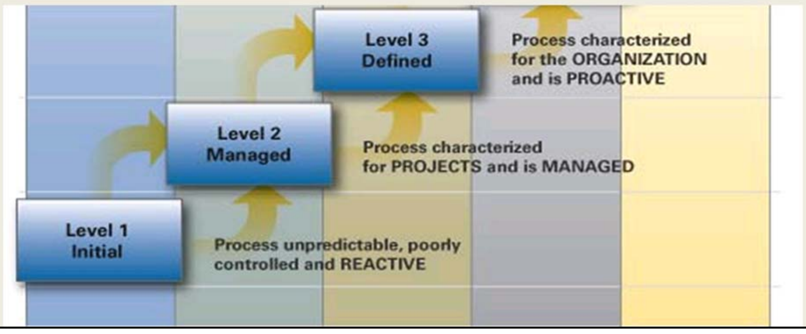
1: All of the specific goals of the process area (defined by CMMI) have been satisfied. Work tasks required to produce defined work products are being conducted.

2: After fulfilling level 1, all work associated with the process area must follow an organizational policy. All people doing the work have access to adequate resources to get the job done; stakeholders are actively involved in the process area as required; all work tasks and work products are “monitored, controlled, and reviewed; and are evaluated for adherence to the process description”;

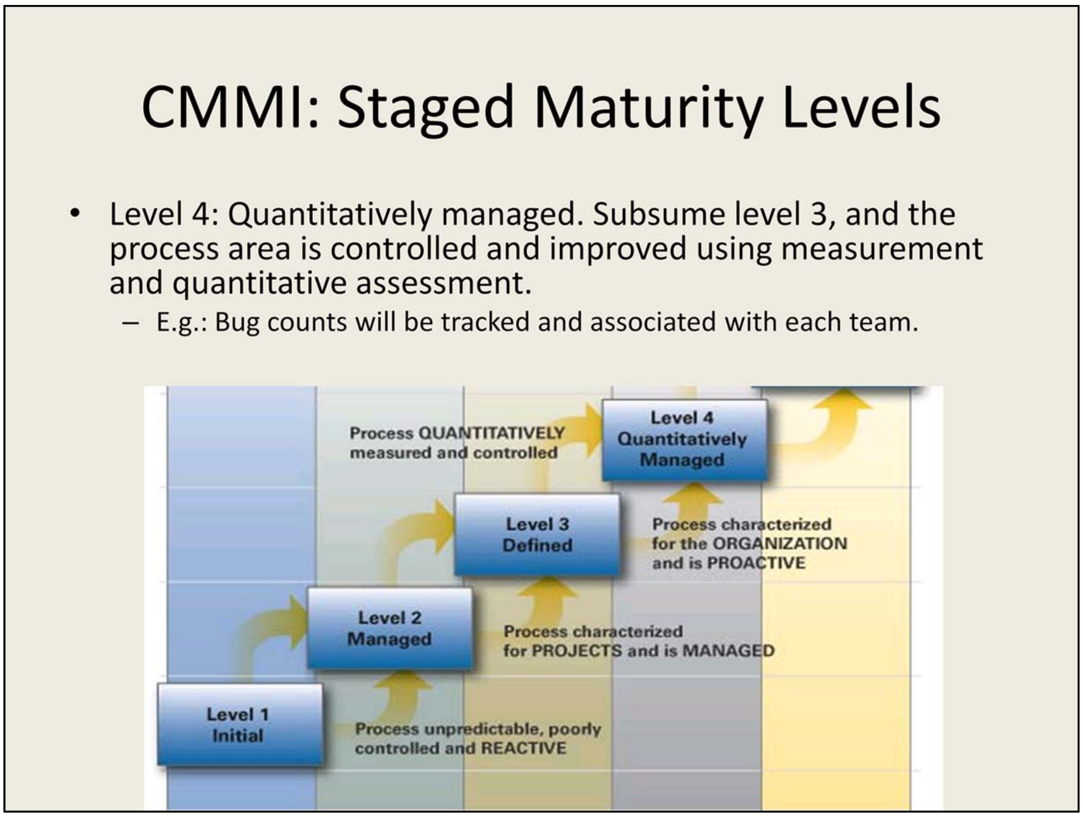


CMMI: Staged Maturity Levels

- Level 3: Defined. Subsume level 2, and the process is systematically tailored to meet organization's need and policy.
  - E.g.: Usability design will be performed in collaboration with usability team to ensure the consistency of user experience across product line.

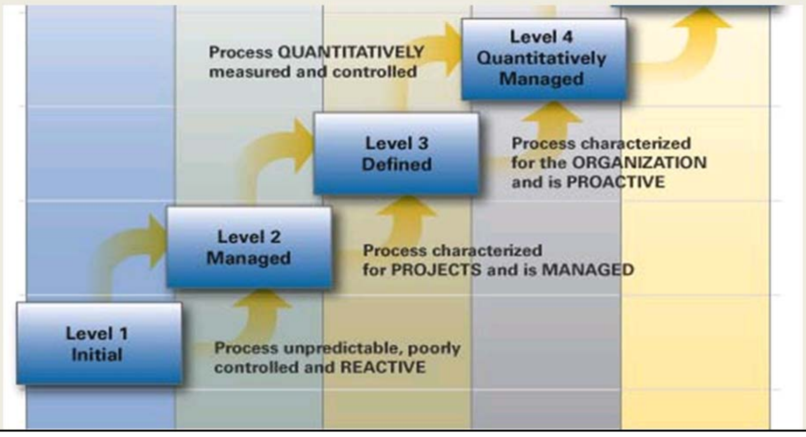

Level 3 Defined
Process characterized for the ORGANIZATION and is PROACTIVE
Level 2 Managed
Process characterized for PROJECTS and is MANAGED
Level 1 Initial
Process unpredictable, poorly controlled and REACTIVE

3: higher level. Corporation level.

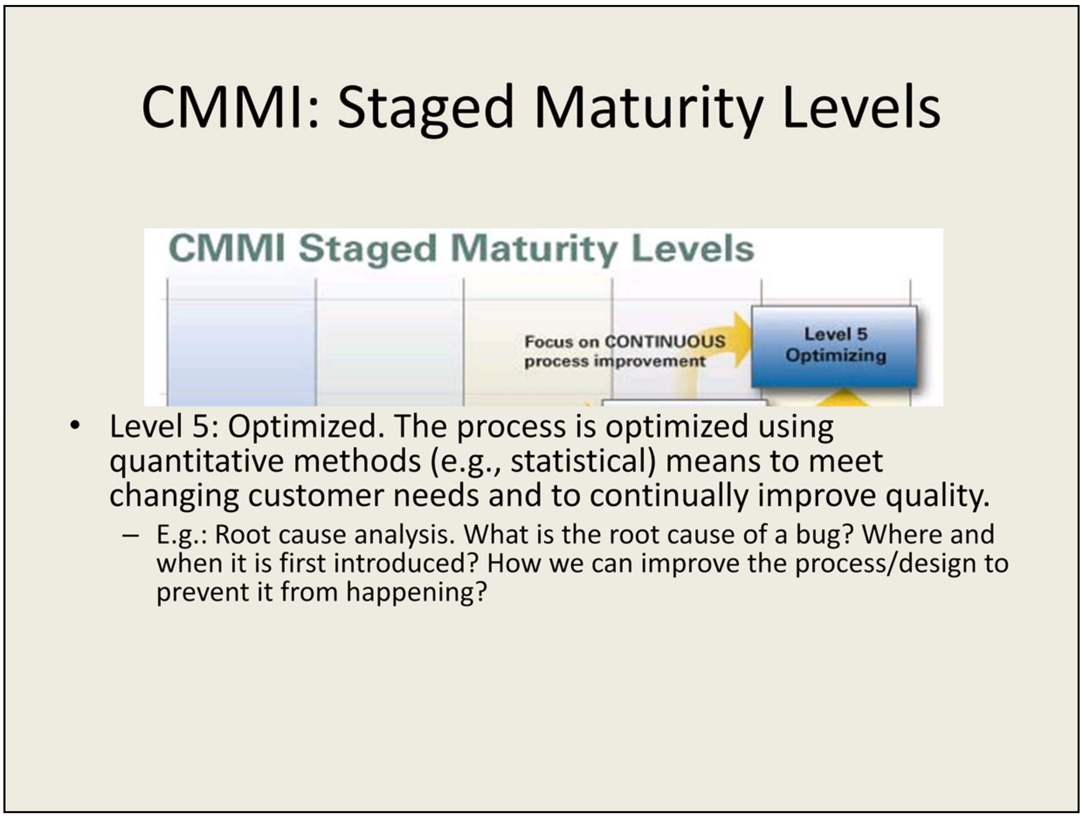


CMMI: Staged Maturity Levels

- Level 4: Quantitatively managed. Subsume level 3, and the process area is controlled and improved using measurement and quantitative assessment.
  - E.g.: Bug counts will be tracked and associated with each team.


Process QUANTITATIVELY measured and controlled
Level 4 Quantitatively Managed
Level 3 Defined
Process characterized for the ORGANIZATION and is PROACTIVE
Level 2 Managed
Process characterized for PROJECTS and is MANAGED
Level 1 Initial
Process unpredictable, poorly controlled and REACTIVE

4: Get down to the numbers.



CMMI: Staged Maturity Levels

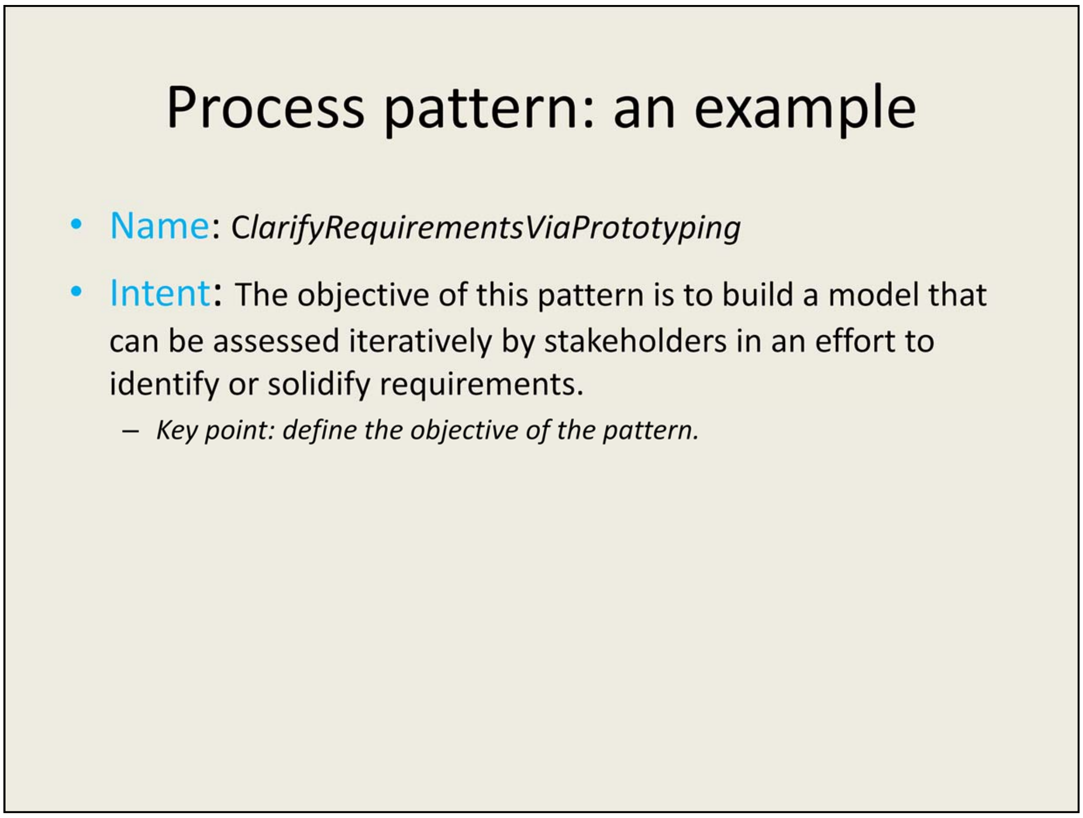
CMMI Staged Maturity Levels
Focus on CONTINUOUS process improvement
Level 5 Optimizing

- Level 5: Optimized. The process is optimized using quantitative methods (e.g., statistical) means to meet changing customer needs and to continually improve quality.
  - E.g.: Root cause analysis. What is the root cause of a bug? Where and when it is first introduced? How we can improve the process/design to prevent it from happening?

5: improve quality and the efficiency.

Process Patterns: Improving process design by reuse

- Process patterns provide a general solution to a common problem/issue in a software process;
  - Promote and distill good practices by defining process patterns;
- Process patterns define the characteristics of a set of activities, actions, work tasks, work products and/or related behaviors. e.g. :
  - Customer communication (a process activity)
  - Project estimating (an action)
  - Requirements gathering (a process task)
  - Reviewing a work product (a process task)
  - Design model (a work product)

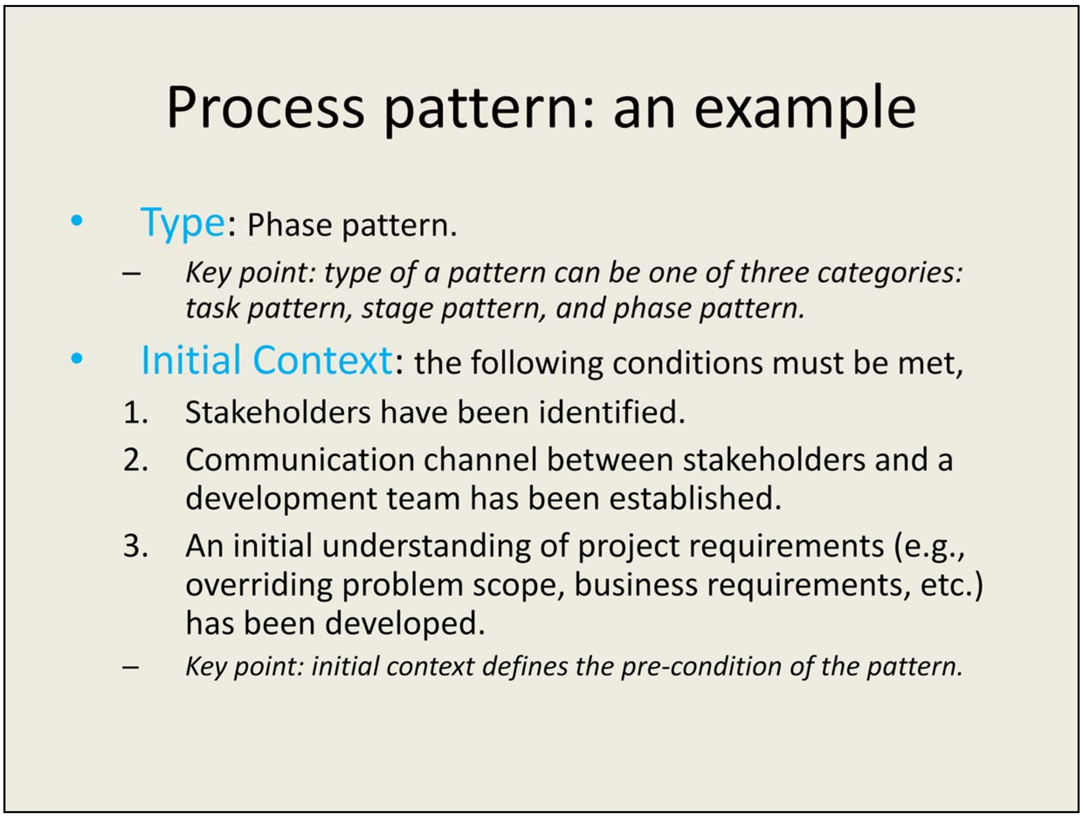


Process pattern: an example

- Name: *ClarifyRequirementsViaPrototyping*
- Intent: The objective of this pattern is to build a model that can be assessed iteratively by stakeholders in an effort to identify or solidify requirements.
  - *Key point: define the objective of the pattern.*

Page 37

Many patterns: analysis, design, testing patterns.



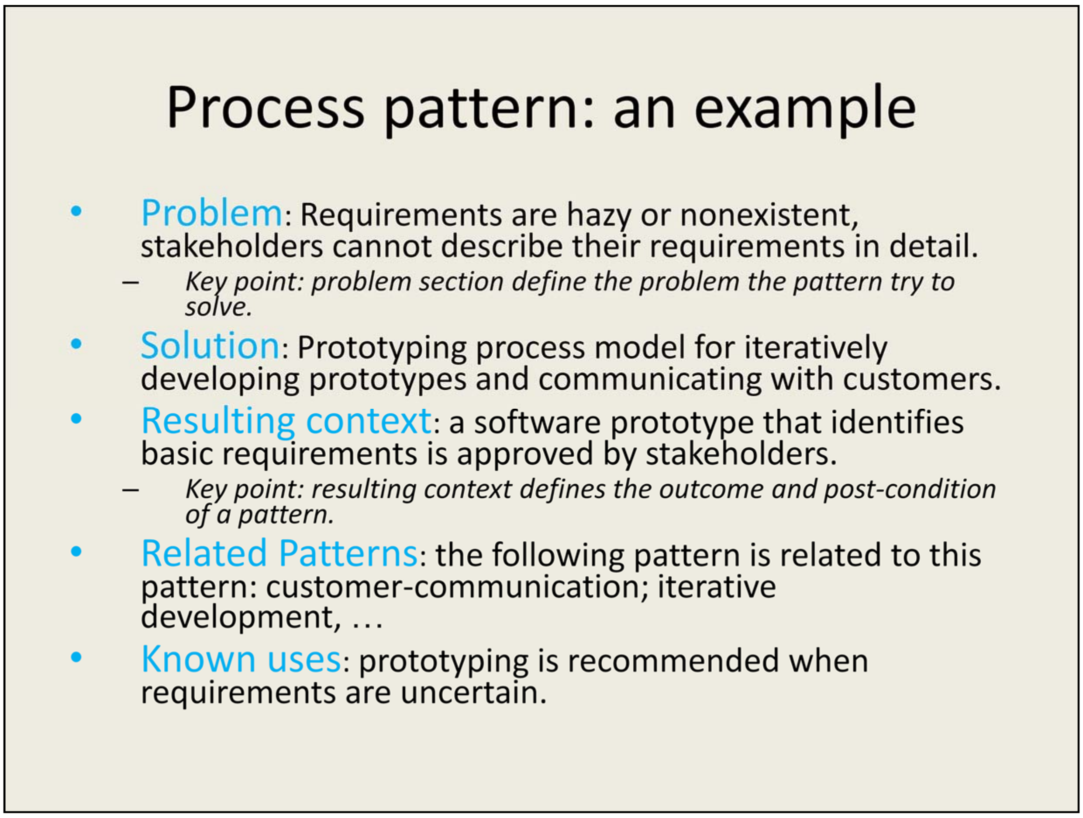
Process pattern: an example

- Type: Phase pattern.
  - *Key point: type of a pattern can be one of three categories: task pattern, stage pattern, and phase pattern.*
- Initial Context: the following conditions must be met,
  1. Stakeholders have been identified.
  2. Communication channel between stakeholders and a development team has been established.
  3. An initial understanding of project requirements (e.g., overriding problem scope, business requirements, etc.) has been developed.
  - *Key point: initial context defines the pre-condition of the pattern.*

Task pattern: software engineering action or work task. E.g. Requirments gathering;

Stage pattern: problem associated with a framework activity, would incorporate multiple task patterns, eg: establishing communication

Phase pattern: the sequence of framework activities – how to organize the frame activities; spiral model / prototyping



Process pattern: an example

- Problem: Requirements are hazy or nonexistent, stakeholders cannot describe their requirements in detail.
  - *Key point: problem section define the problem the pattern try to solve.*
- Solution: Prototyping process model for iteratively developing prototypes and communicating with customers.
- Resulting context: a software prototype that identifies basic requirements is approved by stakeholders.
  - *Key point: resulting context defines the outcome and post-condition of a pattern.*
- Related Patterns: the following pattern is related to this pattern: customer-communication; iterative development, …
- Known uses: prototyping is recommended when requirements are uncertain.

Benefits:

describes the initialte and construct phases of software devlopment

Satisfies the demands of large-scale, mission-critical software

Geared towards the development of business applications.

Provides proven advice based on experience, not academic theory

Indicates how the entire process works together

Software developer for software developer

Straightforward, easy to understand language

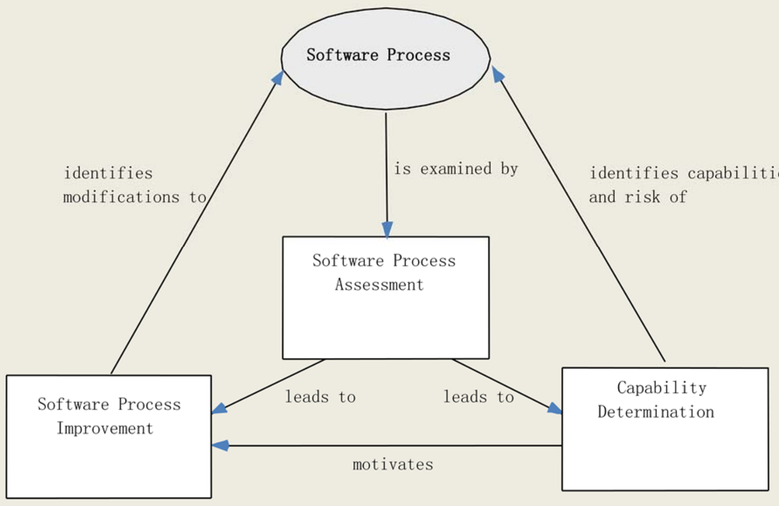
?UML for all models

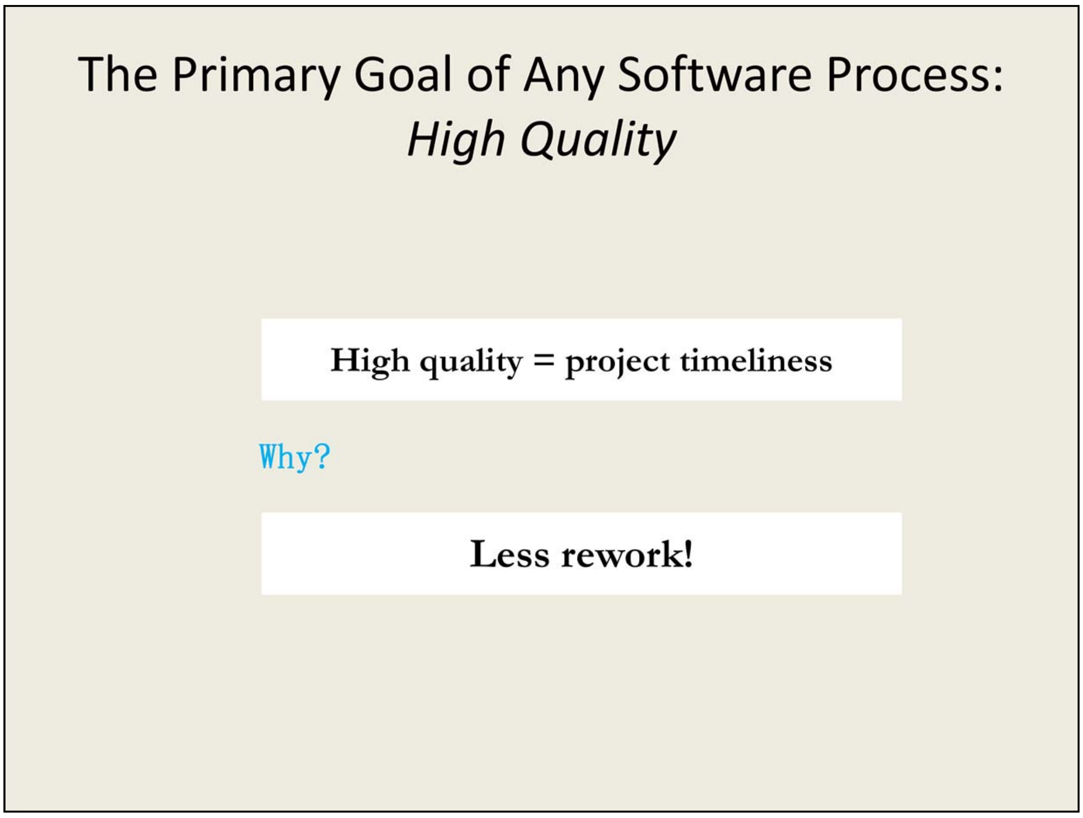
Significantly more comprehensive view of the process. Techniques can be easily adapted.

Process Assessment: Measuring quality of processes

- 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
    - CMMI-based five-step model to assess maturity of development processes.
  - CBA IPI
    - CMM-based appraisal for internal process improvement.
  - SPICE
    - ISO/IEC 15504 defines a set of requirements for software process assessment.
  - ISO 9001:2000
    - General standard for improving quality of product.

Role of Process Assessment


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



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

High quality = project timeliness

Why?

Less rework!

Summary

- Quality focus
- Meta process model
- CMMI
- Process pattern
- Process assessment