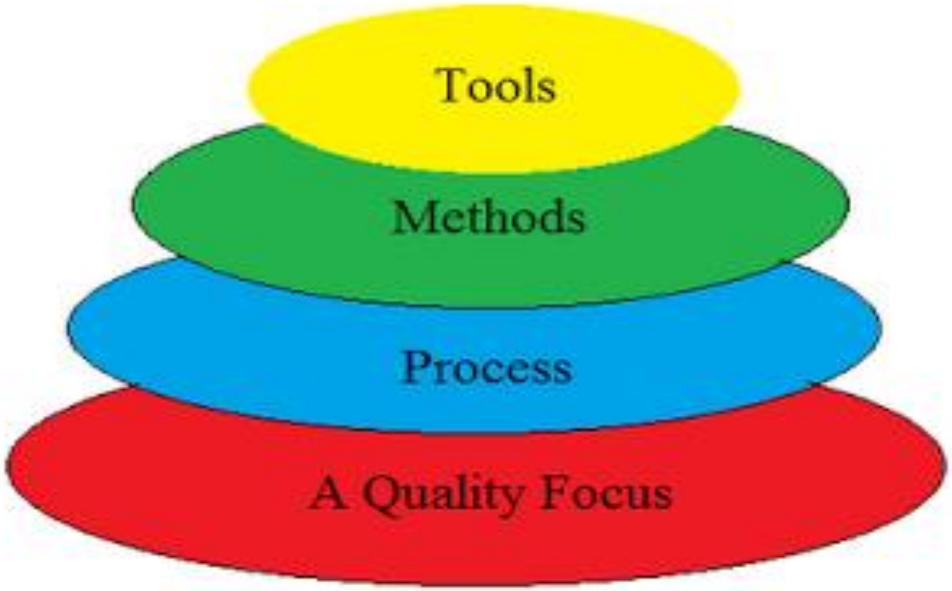
Chapter 2
A Generic View of Process

A Generic view of process

- Software Engineering – A Layered Technology.
- A Process Framework
- The Capability Maturity Model Integration (CMMI),
- Process Patterns,
- Process Assessment
- Personal and Team Process Models,
- Process Technology
- Product and Process

Software Engineering – A Layered Technology.

- Divided into **4** layers



Software Engineering – A Layered Technology.

1. A quality focus :-

Any engineering approach must rest on quality. The "Bed Rock" that supports software Engineering is Quality Focus.

2. Process :-

Foundation for SE is the Process Layer. SE process is the GLUE that holds all the technology layers together and enables the timely development of computer software. It forms the base for management control of software project.(technical methods applied, work products to be produced, milestones to be reached, change is properly managed)

3. Methods :-

SE methods provide the "Technical Questions" for building Software. Methods contain a broad array of tasks that include communication requirement analysis, design modeling, program construction, testing and support.

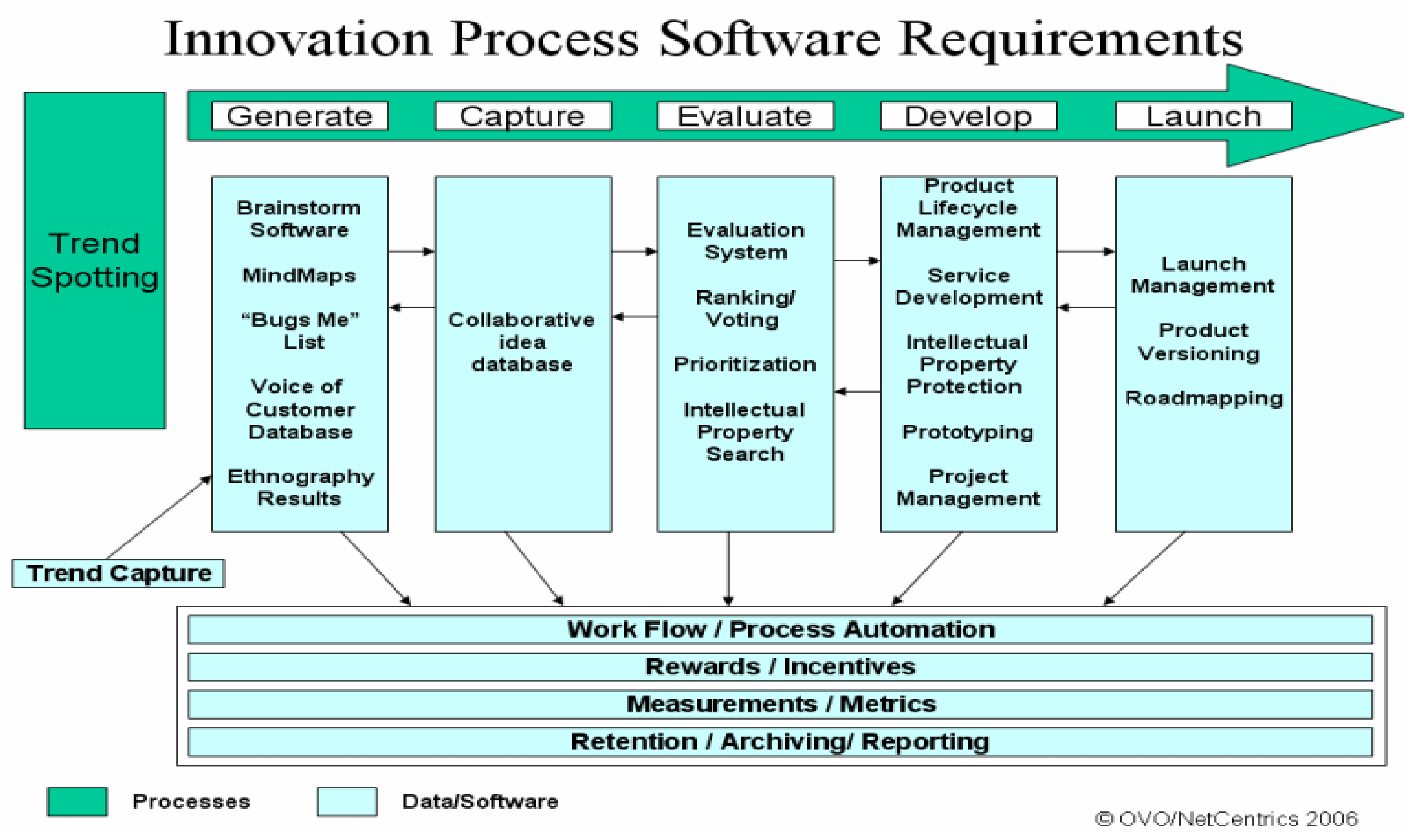
4. Tools :-

SE tools provide automated or semi-automated support for the "Process" and the "Methods". Tools are integrated so that information created by one tool can be used by another.

Software Process framework

- A Process defines, who is doing what, when and how to reach a certain goal…
- Software Process framework is a set of guidelines, concepts and best practices that describes high level processes in software engineering. It does not talk about how these processes are carried out and in what order. Processes like Requirements Analysis, Validation, Verification, Development etc.
- eg Eclipse Process Framework, Microsoft Solutions Framework.

S/W Process framework-Eg.



Innovation Process Software Requirements

Generate → Capture → Evaluate → Develop → Launch

Trend Spotting

Trend Capture

Generate: Brainstorm Software; MindMaps; "Bugs Me" List; Voice of Customer Database; Ethnography Results

Capture: Collaborative idea database

Evaluate: Evaluation System; Ranking/ Voting; Prioritization; Intellectual Property Search

Develop: Product Lifecycle Management; Service Development; Intellectual Property Protection; Prototyping; Project Management

Launch: Launch Management; Product Versioning; Roadmapping

Work Flow / Process Automation

Rewards / Incentives

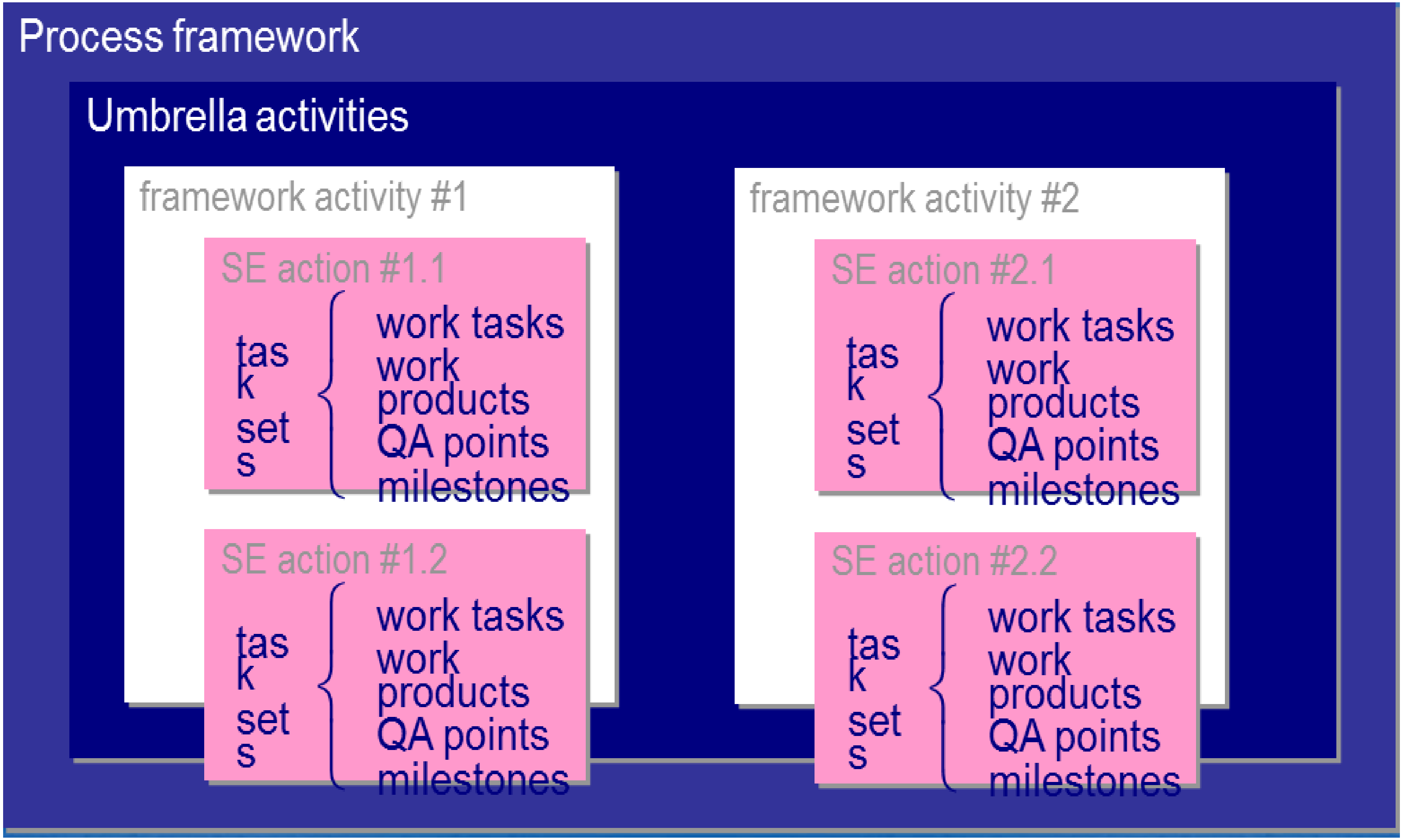
Measurements / Metrics

Retention / Archiving/ Reporting

🟩 Processes

□ Data/Software

© OVO/NetCentrics 2006



Process framework

Umbrella activities

framework activity #1

SE action #1.1

task sets

- work tasks
- work products
- QA points
- milestones

SE action #1.2

task sets

- work tasks
- work products
- QA points
- milestones

framework activity #2

SE action #2.1

task sets

- work tasks
- work products
- QA points
- milestones

SE action #2.2

task sets

- work tasks
- work products
- QA points
- milestones

Generic Process Framework Activities

- Framework activities are applicable to all software projects regardless of size and complexity.
- Each framework activity is populated by a set of software engineering actions (a collection of tasks that produces a work product)
- e.g. requirements gathering is a SE action

Generic Process Framework Activities

1. Communication – lots of communication and collaboration with customer and other stakeholders.. Encompasses requirements gathering.

2. Planning – establishes plan for software engineering work that follows. Describes technical tasks, likely risks, required resources, works products and a work schedule

Generic Process Framework Activities

3. Modeling – encompasses creation of models that allow the developer and customer to better understand software requirements and the design that will achieve those requirements.

4. Construction – code generation and testing.

5. Deployment – software, partial or complete, is delivered to the customer who evaluates it and provides feedback.

Modeling Activity

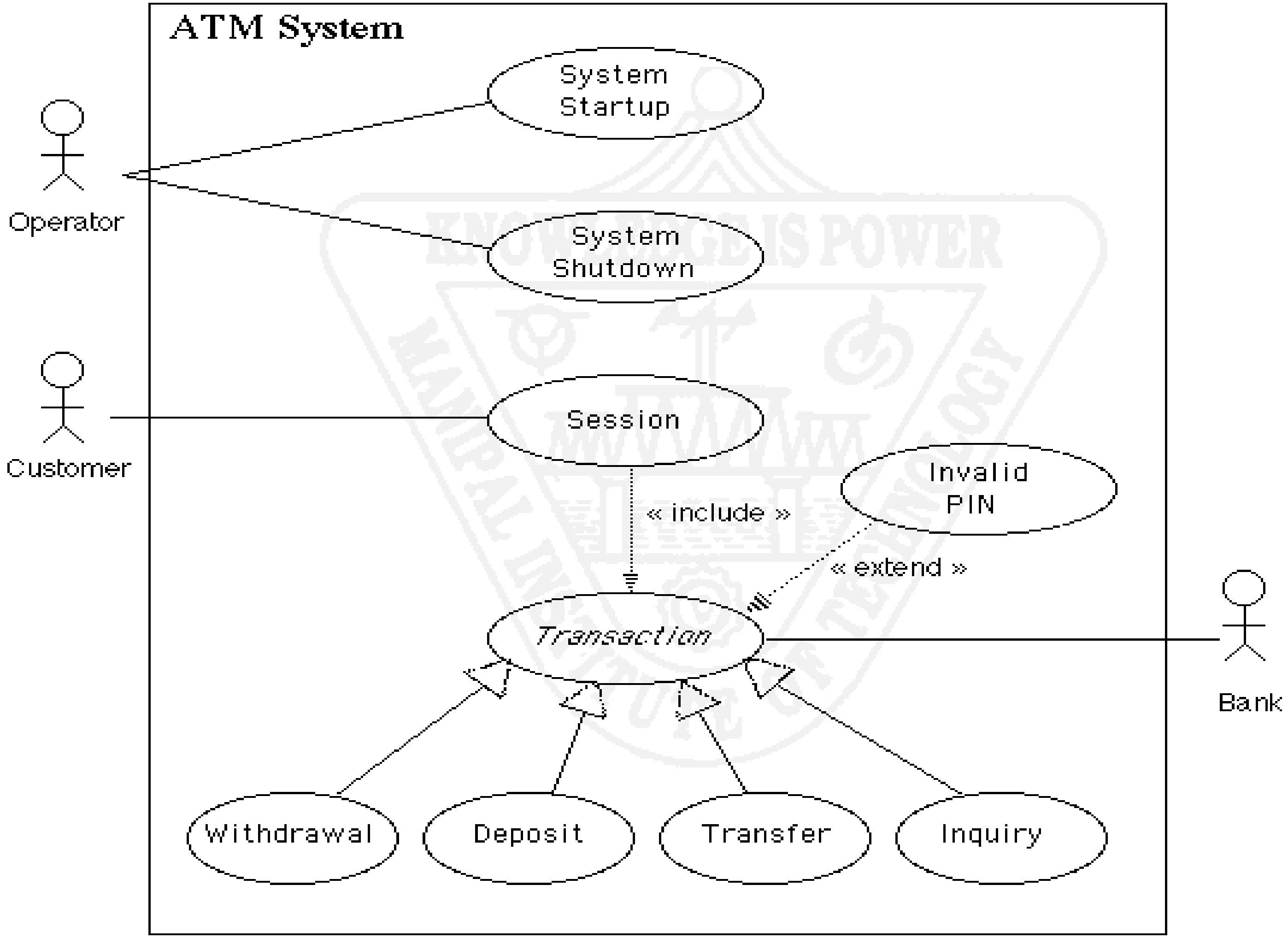
Composed of two software engineering actions

1. analysis – composed of work tasks (e.g. requirement gathering, elaboration, specification and validation) that lead to creation of analysis model and/or requirements specification.

2. design – encompasses work tasks such as data design, architectural design, interface design and component level design. leads to

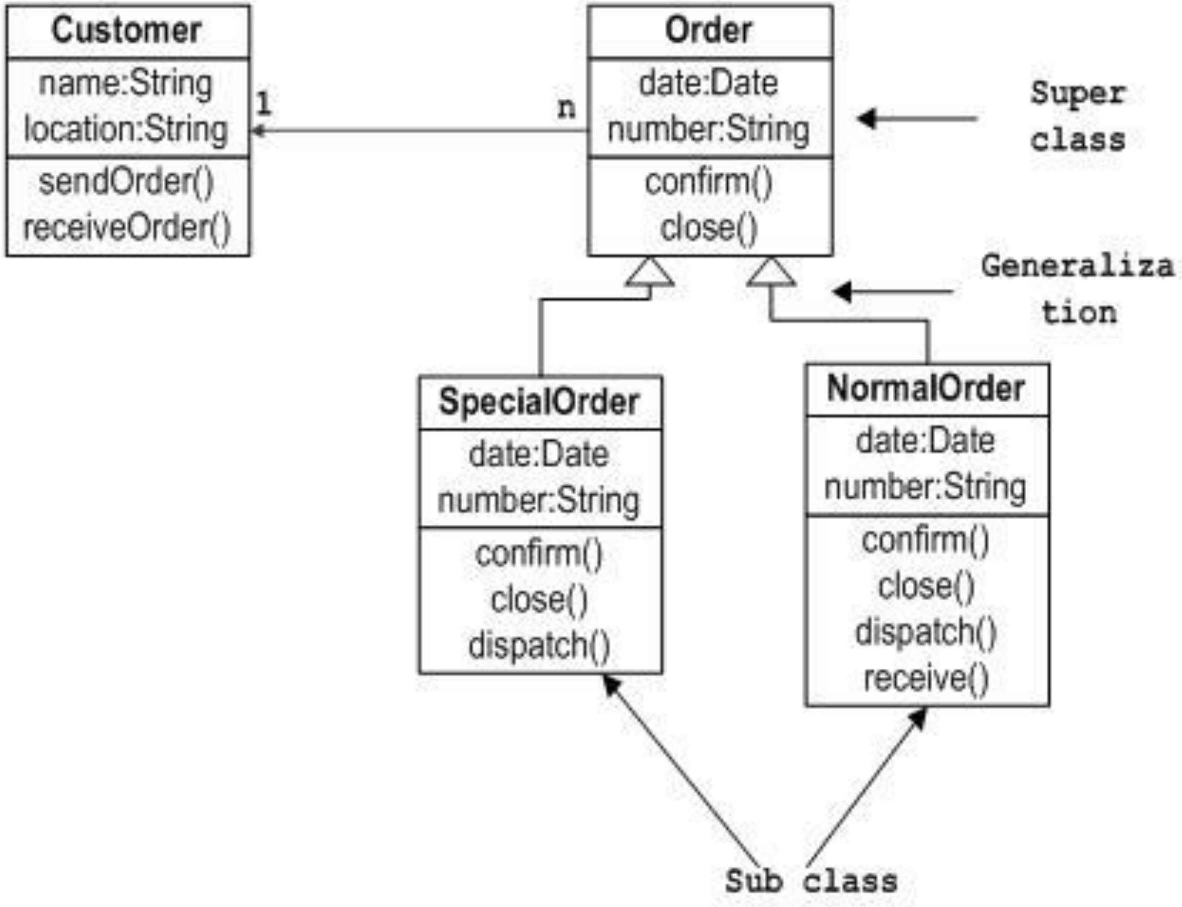
creation of design model and/or a design specification.

Use Case Diagram



Class Diagram

Sample Class Diagram



Project demands..

- Different projects demand different task sets. Software team chooses task set based on problem and project characteristics.
- Requirements gathering is an important software engineering action that occurs in during the communications framework activity. Goal is to understand what various stakeholders want from the software that is to be built.

For a simple project. Requirements Gathering

1. make list of stakeholders
2. invite stakeholders to an informal meeting
3. ask each one to make a list of features and functions required
4. discuss requirements and build a final list
5. prioritize requirements
6. note areas of uncertainty

For a larger projects…

1. make list of stakeholders
2. interview each stakeholder separately to determine overall wants and needs
3. build preliminary list of functions and features based on stakeholder input
4. schedule series of facilitated requirements gathering meetings
5. conduct meetings
6. produce informal user scenarios as part of each meeting
7. refine user scenarios based on feedback
8. build revised list of requirements
9. use quality function deployment to prioritize requirements
10. package requirements so that they can be delivered incrementally
11. note constraints that will be placed on system
12. discuss methods for validating the system

Umbrella Activities

- Framework is augmented by a number of umbrella activities. Typical ones are:

1. Software project tracking and control - allows software team to assess progress against project plan and take necessary action to maintain schedule.

2. Risk management – assess risk that may effect the outcome of the project or the product quality.

Umbrella Activities

3. Formal technical reviews – assess software engineering work products to uncover and remove errors before they are propagated to the next action or activity.

4. Measurement – defines and collects process, project and product measures that assist team in developing software

5. Software configuration management – manages the effect of change throughout the software process

Umbrella Activities

- Reusability management – defines criteria for work product reuse and establishes mechanism to achieve reusable components
- Work product preparation and production – included work activities required to create work products such as documents, logs, models.

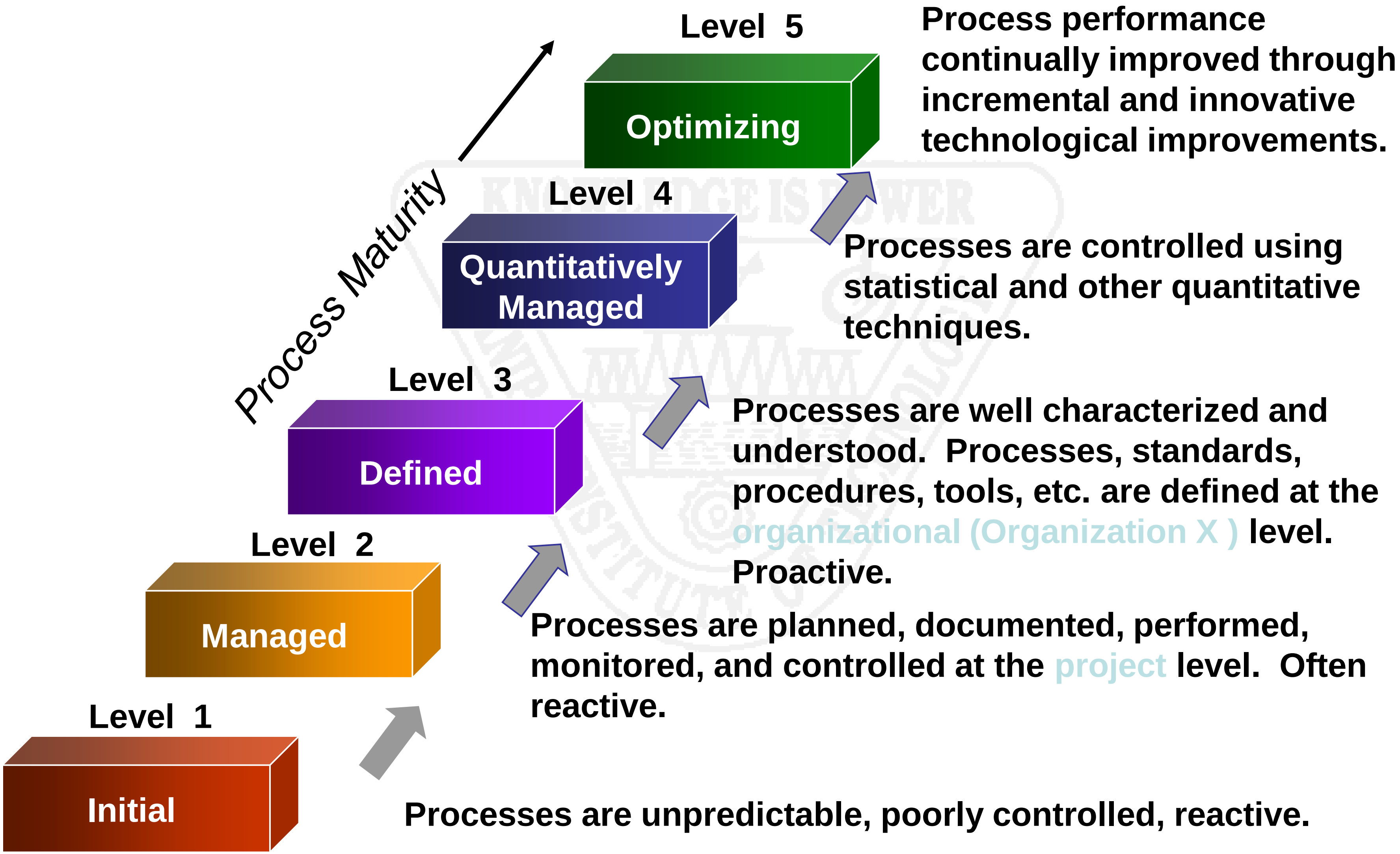
Process models differ in…??

- overall flow of activities and interdependencies among activities and task
- the degree to which tasks are defined
- the degree to which work products and identified and required
- the way in which QA activities are applied
- the way in which project tracking and control is done

Capability Maturity Model Integration (CMMI)

- CMMI (Capability Maturity Model Integration) is a proven industry framework to improve product quality and development efficiency for both hardware and software
  - Sponsored by US Department of Defence in cooperation with Carnegie Mellon University and the Software Engineering Institute (SEI)
  - Many companies have been involved in CMMI definition such as Motorola and Ericsson
  - CMMI has been established as a model to improve business results
- CMMI, staged, uses 5 levels to describe the maturity of the organization, same as predecessor CMM
  - Vastly improved version of the CMM
  - Emphasis on business needs, integration and institutionalization

CMMI Staged Representation - 5 Maturity Levels



Process performance continually improved through incremental and innovative technological improvements.

Processes are controlled using statistical and other quantitative techniques.

Processes are well characterized and understood. Processes, standards, procedures, tools, etc. are defined at the organizational (Organization X) level. Proactive.

Processes are planned, documented, performed, monitored, and controlled at the project level. Often reactive.

Processes are unpredictable, poorly controlled, reactive.

Maturity Level 2
Managed at the Project Level

- Maturity Level 2 deals with managed processes.
- A managed process is a performed process that is also:
  - Planned and executed in accordance with policy
  - Employs skilled people
  - Adequate resources are available
  - Controlled outputs are produced
  - Stakeholders are involved
  - The process is reviewed and evaluated for adherence to requirements
- Processes are planned, documented, performed, monitored, and controlled at the project level. Often reactive.
- The managed process comes closer to achieving the specific objectives such as quality, cost, and schedule.

Behaviors at the Five Levels

Maturity Level	Process Characteristics	Behaviors
5 Optimizing	Focus is on continuous quantitative improvement	Focus on "fire prevention"; improvement anticipated and desired, and impacts assessed.
4 Quantitatively Managed	Process is measured and controlled	Greater sense of teamwork and inter-dependencies
3 Defined	Process is characterized for the organization and is proactive	Reliance on defined process. People understand, support and follow the process.
2 Managed	Process is characterized for projects and is often reactive	Over reliance on experience of good people – when they go, the process goes. “Heroics.”
1 Initial	Process is unpredictable, poorly controlled, and reactive	Focus on "fire fighting"; effectiveness low – frustration high.

Process Assessment and Improvement

- **Standard CMMI Assessment Method for Process Improvement (SCAMPI)** — provides a five step process assessment model that incorporates five phases: initiating, diagnosing, establishing, acting and learning.
- **CMM-Based Appraisal for Internal Process Improvement (CBA IPI)**—provides a diagnostic technique for assessing the relative maturity of a software organization; uses the SEI CMM as the basis for the assessment [Dun01]
- **SPICE—The SPICE (ISO/IEC15504)** standard defines a set of requirements for software process assessment. The intent of the standard is to assist organizations in developing an objective evaluation of the efficacy of any defined software process. [ISO08] etc..

Personal Software Process (PSP)

- **Planning.** This activity isolates requirements and develops both size and resource estimates. In addition, a defect estimate (the number of defects projected for the work) is made. All metrics are recorded on worksheets or templates. Finally, development tasks are identified and a project schedule is created.
- **High-level design.** External specifications for each component to be constructed are developed and a component design is created. Prototypes are built when uncertainty exists. All issues are recorded and tracked.
- **High-level design review.** Formal verification methods are applied to uncover errors in the design. Metrics are maintained for all important tasks and work results.
- **Development.** The component level design is refined and reviewed. Code is generated, reviewed, compiled, and tested. Metrics are maintained for all important tasks and work results.
- **Postmortem.** Using the measures and metrics collected (this is a substantial amount of data that should be analyzed statistically), the effectiveness of the process is determined. Measures and metrics should provide guidance for modifying the process to improve its effectiveness.

Team Software Process (TSP)

- Build self-directed teams that plan and track their work, establish goals, and own their processes and plans. These can be pure software teams or integrated product teams (IPT) of 3 to about 20 engineers.
- Show managers how to coach and motivate their teams and how to help them sustain peak performance.
- Accelerate software process improvement by making CMM Level 5 behavior normal and expected.
  - The Capability Maturity Model (CMM), a measure of the effectiveness of a software process.
- Provide improvement guidance to high-maturity organizations.
- Facilitate university teaching of industrial-grade team skills.

Process technology

- Process technology tools have been developed to help software organizations to analyze their current process, organize work tasks, control and monitor progress and manage technical quality.
- Process technology tools are used to represent key elements of a process so that it can be better understood
- Help to understand the actions and work tasks that are required to perform the process
- Eg: Igrafx Process Tools by Corel Corporation

Product and process..

- If the process is weak, the end product will undoubtedly suffer.
- But an obsessive over reliance on process is also dangerous.
- A creative software developer should derive as much satisfaction from the process as the end product.