1.264 Lecture 4

Unified Modeling Language (UML)
Software Process: CMM, ISO

Unified Modeling Language

- Object-oriented modeling language, migrated from relational database modeling
  - Standard managed by Object Management Group
  - Many implementations of UML (Microsoft, IBM/Rational, Borland/Inprise)
- Combines previously competing approaches
  - Rumbaugh Object Modeling Technique (OMT)
  - Shlaer-Mellor method
  - Booch method
- Substantial level of use currently, becoming common
  - More common in J2EE than in .Net, though .Net use increasing sharply also

Unified Modeling Language, p.2

- Why is UML coming into wider use?
  - Speed up requirements process
  - Lessen information loss between requirements and design processes, and between design and implementation
  - Communication: clearer than natural language, provides a level of precision, but avoids details
  - Supports iterative development (i.e., spiral model)
    - Supports both high level requirements/design in early spirals and detailed requirements/design later
- UML is just the modeling language
  - Rational Unified Process (RUP) is a recommended process, based on using UML
    - Inception (requirements)
    - Elaboration (design)
    - Construction (development): “extreme programming” fashionable
    - Transition (test, implementation)

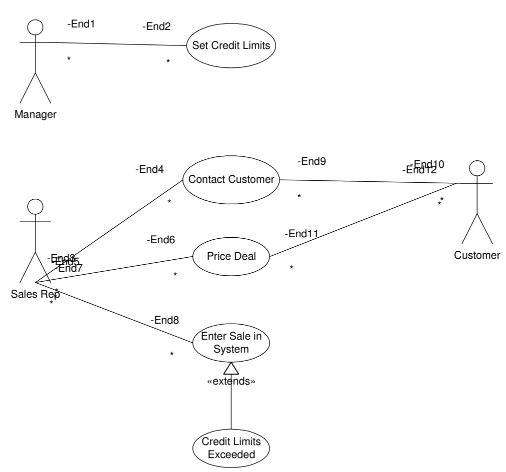
Unified Modeling Language, p.3

- Used in requirements:
  - Deployment diagram, component diagram to show high level view of system
  - Use cases, which are very structured scenarios used to define system requirements
    - Good basic approach, but needs narrative to support
- Used in design:
  - Data models (not strictly part of UML) are done in conjunction with class (object) models and correspond closely
    - Often done area by area and then consolidated
  - Activity diagrams, used to model workflows, to find duplicate processes that can be eliminated
  - Prototyping used for risky, critical, difficult parts of system

UML static model diagrams

- **Use case diagram**
  - **Drawings and structured descriptions of steps in workflows**
- **Class diagram**
  - **Internal structure of system, extension of entity-relationship diagram.**
  - **Three elements in each entity: name, attributes, methods**
- **Deployment diagram**
  - **Physical components: processors, workstations, network**
- **Package and/or component diagram**
  - **High level model of physical software architecture**
  - **Consists of modules, which are grouped in packages**
  - **Package contains definition of group of classes (entities, methods)**

Use case example

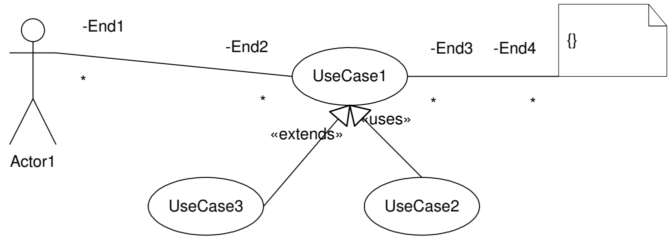


Use case exercise

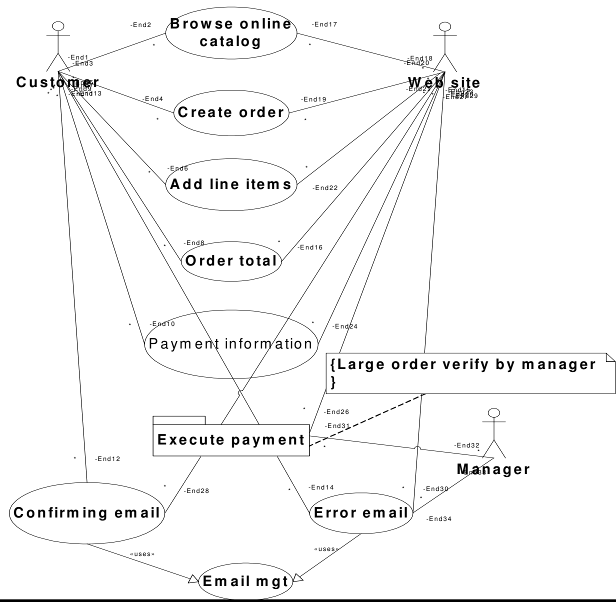
- Customer browses online catalog of chemical products and adds desired products as line items to order.
- When order is complete, customer provides credit card or purchase order number
- System checks credit card or purchase order information and sends confirmation email
- For large orders, manager verifies payment
- Exercise:
  - Create a use case diagram in Visio for this case
  - Optionally, begin to create a text version of this use case with more detail

Exercise

- If you have Visio on your laptop, please start it now, select Software->UML -> UML Use Case
- If you don't have Visio, please do the exercise on paper
- Key symbols



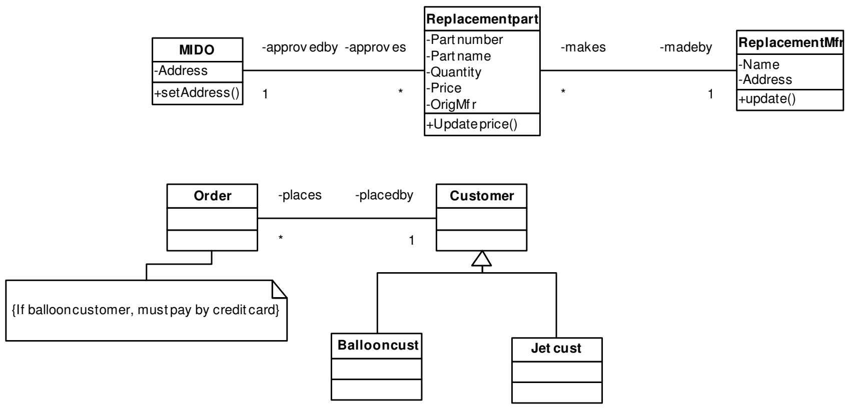
Use case exercise



Class Diagrams

- Used in requirements, design and implementation:
  - Conceptual, to represent general entities in system
  - Specification, where we specify what each entity (class) will do (but not how)
    - List the methods/actions
  - Implementation
    - Detailed class diagram of actual software (Java or C++)
- List attributes, same as data model
- List methods/operations/functions
  - Activities naturally associated with the data in the entity
- Also model constraints, preconditions, postconditions, etc. that are laid out in the use cases
- We often don't model everything—too hard to read
  - Focus on key parts of system

Class Diagram

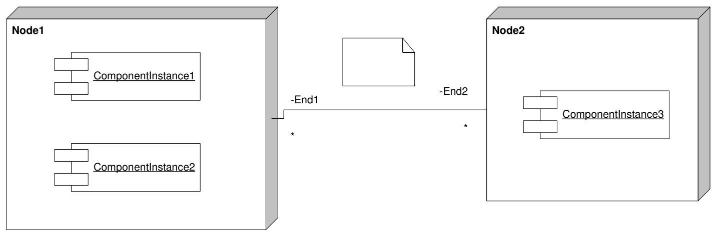


Deployment diagram exercise

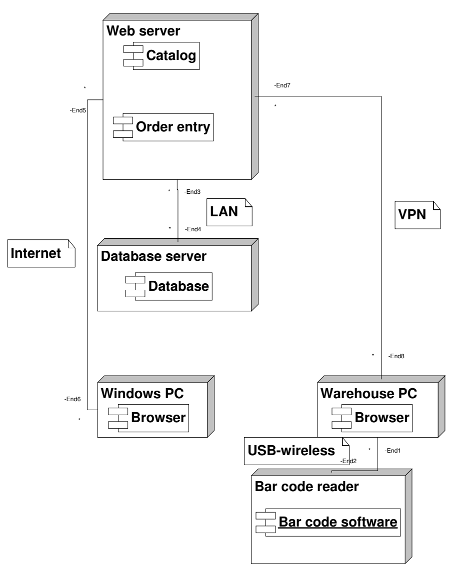
- Chemical company has a Web server on which the catalog and order entry are available
  - There is a separate server that runs the database
- Warehouses have PCs with browsers and bar code scanners to verify inbound inventory and outbound orders
- Users have PCs with browsers
- All machines connected via Internet (tcp/ip)
- Exercise:
  - Draw the deployment diagram using Visio

Exercise

- If you have Visio, please select UML Deployment
- If you don't please do the exercise on paper. Key symbols:



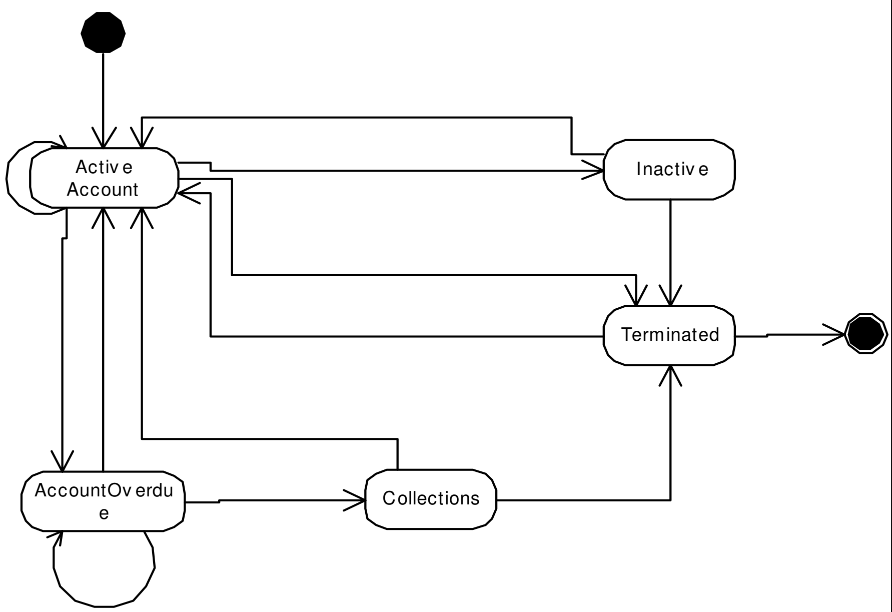
Deployment diagram



Dynamic models

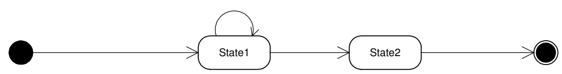
- While static models are done for the system as a whole, dynamic models are done only for key components
- State diagram
  - Specifies behavior of an object (entity)
- Sequence diagram (or ladder diagram)
  - Shows details of scenario and messages that flow between objects over time
  - Heavily used in standards
- Collaboration diagram
  - Shows flow of messages as a graph.

State diagram

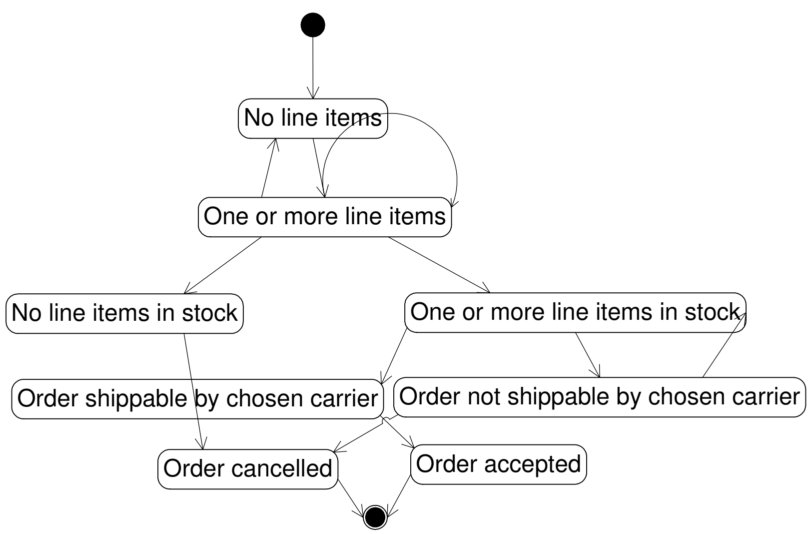


State diagram exercise

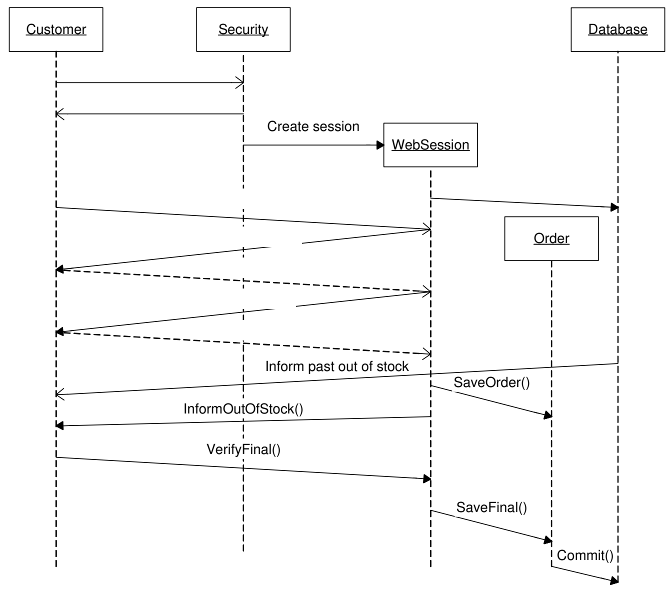
- Chemical company Web site takes orders, line by line, and confirms/denies that product is in stock, either line by line or at end of order
- At end of order, Web site requests preferred mode and carrier and confirms/denies whether shipment can be made on that mode/carrier
- Model the states that the order may have



State diagram



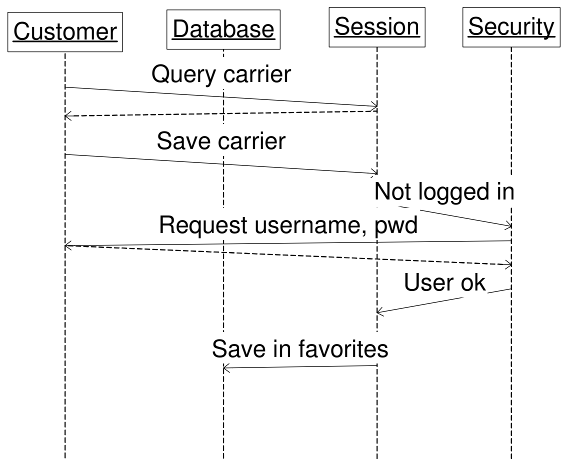
Sequence diagram



Sequence diagram exercise

- Chemical company Web site lists transportation modes and carriers. User can, at any time, create a ‘favorites’ list of carriers, but will be prompted to log in if she hasn’t done so

Sequence diagram



UML Summary

- **Use UML after:**
  - **Writing scenarios and narratives as an initial requirements document**
    - **Refine them into use cases**
  - **Preparing the initial data model**
    - **Add operations/methods to the entities, after understanding the data, to create a class diagram**
- **Use UML deployment, package and/or component models to give overview of the system, in requirements**
- **Use UML use cases to specify processes**
- **Use UML state, collaboration, and/or sequence models selectively in complex parts of the system**
- **UML is becoming a 'universal' language: new staff coming to a project can read it, and this reduces the learning curve very substantially**
  - **Developers and analysts can both understand it readily**
  - **I use UML even for analysis-only projects (as well as writing requirements and modeling data)**

Capability Maturity Model for Software

- Developed at Software Engineering Institute (SEI), Carnegie-Mellon University (www.sei.cmu.edu)
- De facto standard for software process assessment
- Five level model
  - 1: Initial
  - 2: Repeatable
  - 3: Defined
  - 4: Managed
  - 5: Optimized
- Predictability, effectiveness and control of software processes improve as organization moves up these levels

CMM Motivation

- 20 years of unfulfilled promises about productivity and quality gains from new software technology
- Organizations realized fundamental problem is the inability to manage the software process
- CMM provides guidance on how to evolve toward a culture of software engineering and rational management

CMM Level 1: Initial

- Ad hoc, occasionally chaotic
- Few processes defined
- Success depends on individual effort and heroics

CMM Level 2: Repeatable

- Basic project management processes established to track cost, schedule, functionality
- Discipline in place to repeat earlier successes on projects with similar applications
- Key processes focus on basic project management controls
  - Requirements management: initial spec, change control
  - Software project planning: resource estimation based on past performance
  - Software project tracking and oversight
  - Software subcontract management
  - Software quality assurance: code reviews, test plans, tracking
  - Software configuration management
- At level 2, you can measure what’s going on, and that helps understand future projects

CMM Level 3: Defined

- Software process for management and development is documented, standardized and integrated into an overall process for the organization
- All projects use approved, tailored version of standard process
- Key process areas focus on institutionalizing effective process
  - Organization process: spiral model, tracking, control
  - Training program: managers, analysts, developers
  - Integrated software management: lines of code, defects, time,…
  - Software product engineering: alternative designs, scopes…
  - Intergroup coordination: Web, database, analyst, coder, customer
  - Peer reviews: initiate at requirements, design, QA. Continue code reviews
- At level 3, you begin to have some control; you can actually predict times/costs for new projects and make some choices on how to approach them (fastest, most efficient, nominal…)

CMM Level 4: Managed

- Detailed measures of software process and product quality are collected
- Process and products are quantitatively understood and controlled
- Key processes focus on quantitative understanding of process
  - Quantitative process management: measure and manage time at all stages: requirements, design, code, QA
  - Software quality management: reviews at all stages, tests and traceability at all stages
- At level 4 you have real control: you can measure and manage all aspects of the project

CMM Level 5: Optimizing

- Continuous process improvement through quantitative feedback
- Piloting innovative technology and ideas
- Key process areas focus on continual process improvement
  - Defect prevention ('extreme programming', refactoring, reviews)
  - Technology change management (new tools, training, pilots, tests, knowledge sharing)
  - Process change management
- At level 5, you not only have control but are efficient

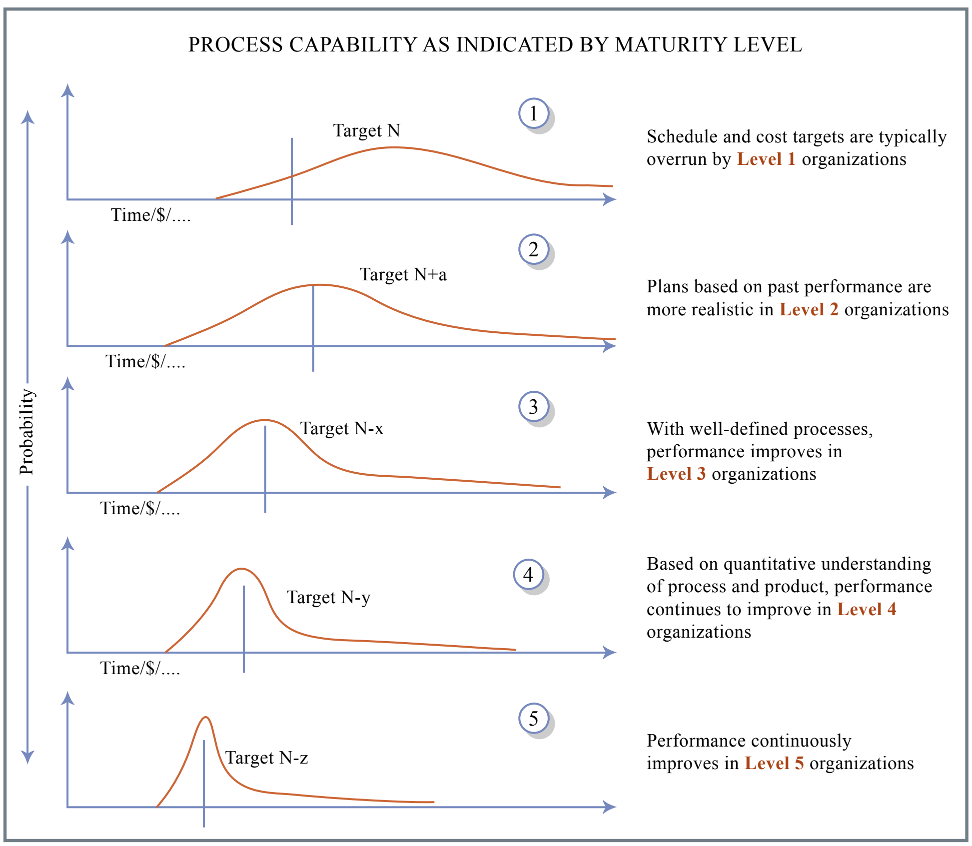
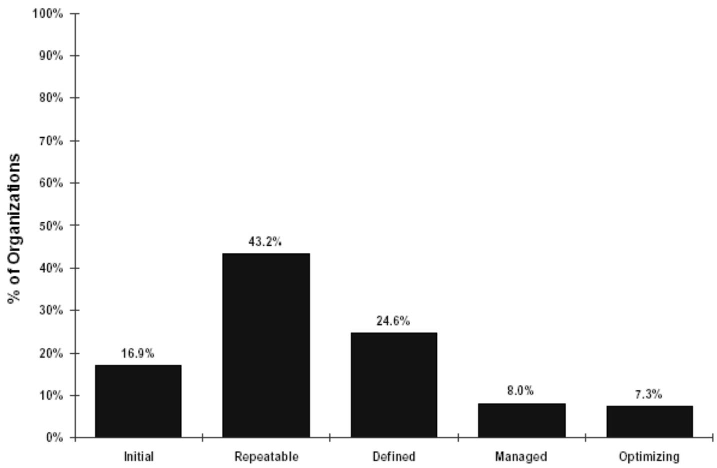


Figure by MIT OCW.

Carnegie Mellon University
Software Engineering Institute

Software CMM - CBA IPI and SPA Appraisal Results

Organization Maturity Profile

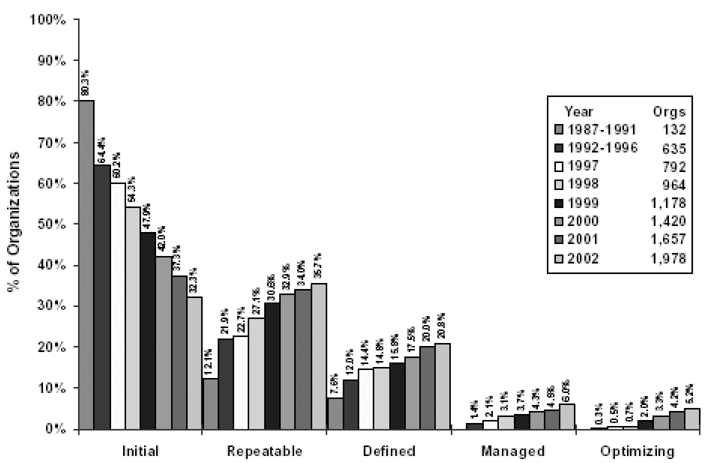


Based on most recent appraisal, since **1998**, of **1,345** organizations

Carnegie Mellon University
Software Engineering Institute

Software CMM - CBA IPI and SPA Appraisal Results

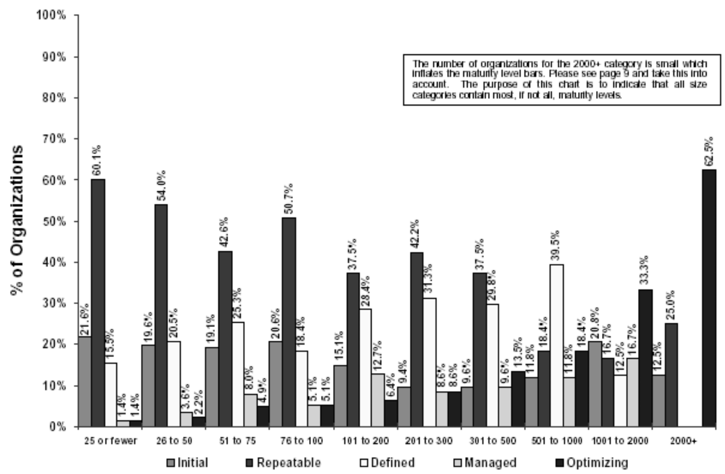
Trends in the Community Maturity Profile



Based on a cumulative view of the most recent appraisals of organizations up through the year indicated.
This accounts for the difference from the figures on page 10.

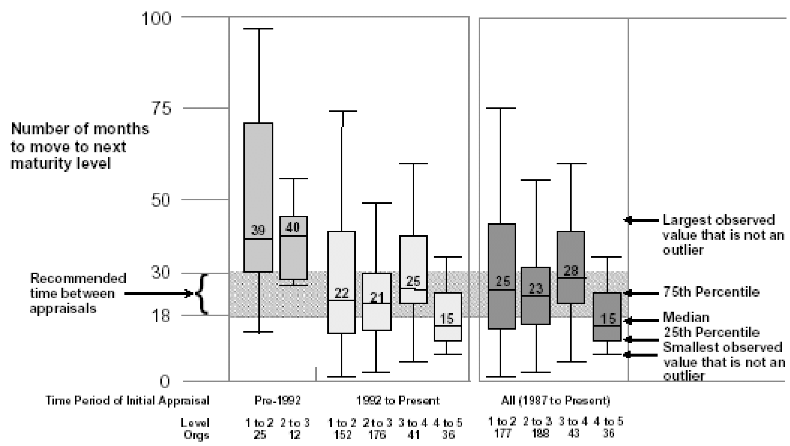
Maturity Profile by Organization Size

Based on the total number of employees within the area of the organization that was appraised



Based on **1,309** organizations reporting size data

Time to Move Up



ISO 9000

- International Standards Organization (ISO)
  - National standards bodies from 100+ countries
- ISO 9000
  - Family of QA standards for framework, models, specifications for quality management systems
  - Best practices applied to production
  - These are process standards, not product standards
- ISO 9001 presents requirements
  - QA standards for design, development and service organizations
  - Approximately 20 process requirements must be met
  - Documenting and standardizing processes to develop end product
  - ISO 9001 does not standardize products
- ISO 9000 and ISO 9004 present guidelines
  - ISO 9002 and ISO 9003 have been dropped

ISO 9000 cont

- ISO 9000 certification mandatory to do business in Europe, and becoming so in Pacific Rim and eventually Americas
- General approach
  - Develop a quality team
  - Say what you do: document processes, usually via flow diagrams
  - Do what you say
  - Prove it: Independent firm (registrar) audits your organization and grants certification
- Often, no real change resulted from ISO 9000
  - ISO 9000 2000 does generally require change, and is mandatory now
  - Gap analysis done for companies with previous QA plan
  - System development plans exist for companies without