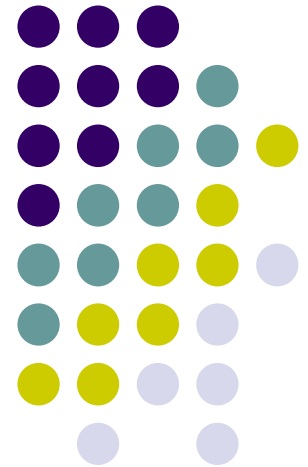
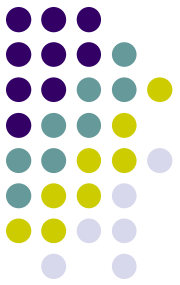


# Characterizing the Software Process

---

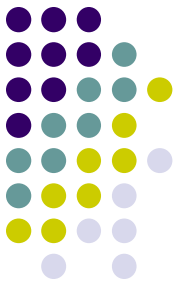
A Maturity Framework  
By: Watts S. Humphrey





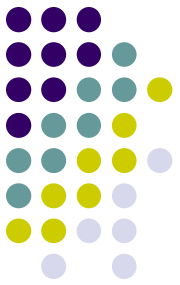
# Background

- Software is a major and increasing portion of US Defense Dept. costs, and software often affects the schedules and effectiveness of weapons systems.
- The Defense Dept. formed the Software Engineering Institute at Carnegie Mellon University in 1984 to establish standards of excellence for software engineering and to accelerate the transition of advanced technology and methods into practice.
- One SEI project is to provide the Defense Dept. with some way to characterize the capabilities of software-development organizations.
- The result is the software-process maturity framework, which can be used by any software organization to assess its own capabilities and identify the most important areas for improvement.



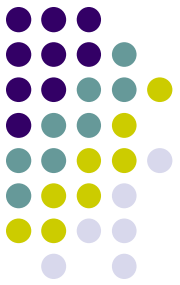
# Idea software process

- First, it is predictable: Cost estimates and schedule commitments are met with reasonable consistency and the quality of the resulting products generally meet user needs.



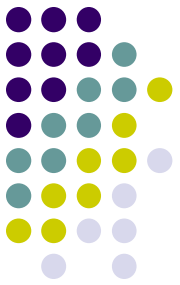
# What is Statistical control

- We define “in statistical control” as follows: A process is in statistical control if and only if measurements made on the process are close enough to those predicted by the model for the process.



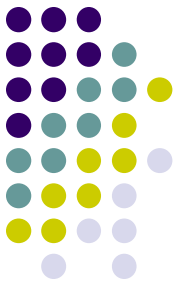
# Statistical control

- The basic principle of software process management is that if the development process is under statistical control, a consistently better result can be achieved only by improving the process.
- If the process is not under statistical control, sustained progress is not possible until it is. Experience from Japanese Industry.
- When a process is under statistical control, repeating the work in roughly the same way will produce roughly the same result.



# Measurement

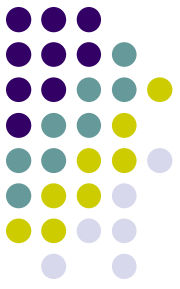
- The basic principle behind statistical control is measurement.
- As Lord Kelvin said a century ago, “...when you can measure what you are speaking about, and express it in numbers, you know something about it; but when you cannot measure it, when you can not express it in numbers, your knowledge is of a meager and unsatisfactory kind; it may be the beginning of knowledge, but you have scarcely in your thoughts advanced to the stage of science....”



# Several factors

- Perhaps most important is that the mere act of measuring human processes changes them. Since people's fears and motivations are involved, the results must be viewed in a different light than data on natural phenomena.
- It is also essential to limit the measurements to those few items that will really be used. Measurement are both expensive and disruptive.

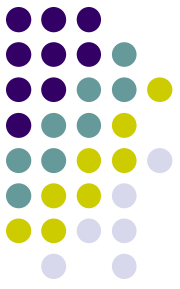
# Development-process improvement



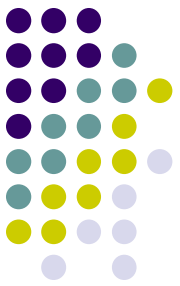
- An important first step in addressing software problems is to treat the entire development task as a process that can be controlled, measured, and improved.
- We define a process as a sequence of tasks that, when properly performed, produces the desired result.



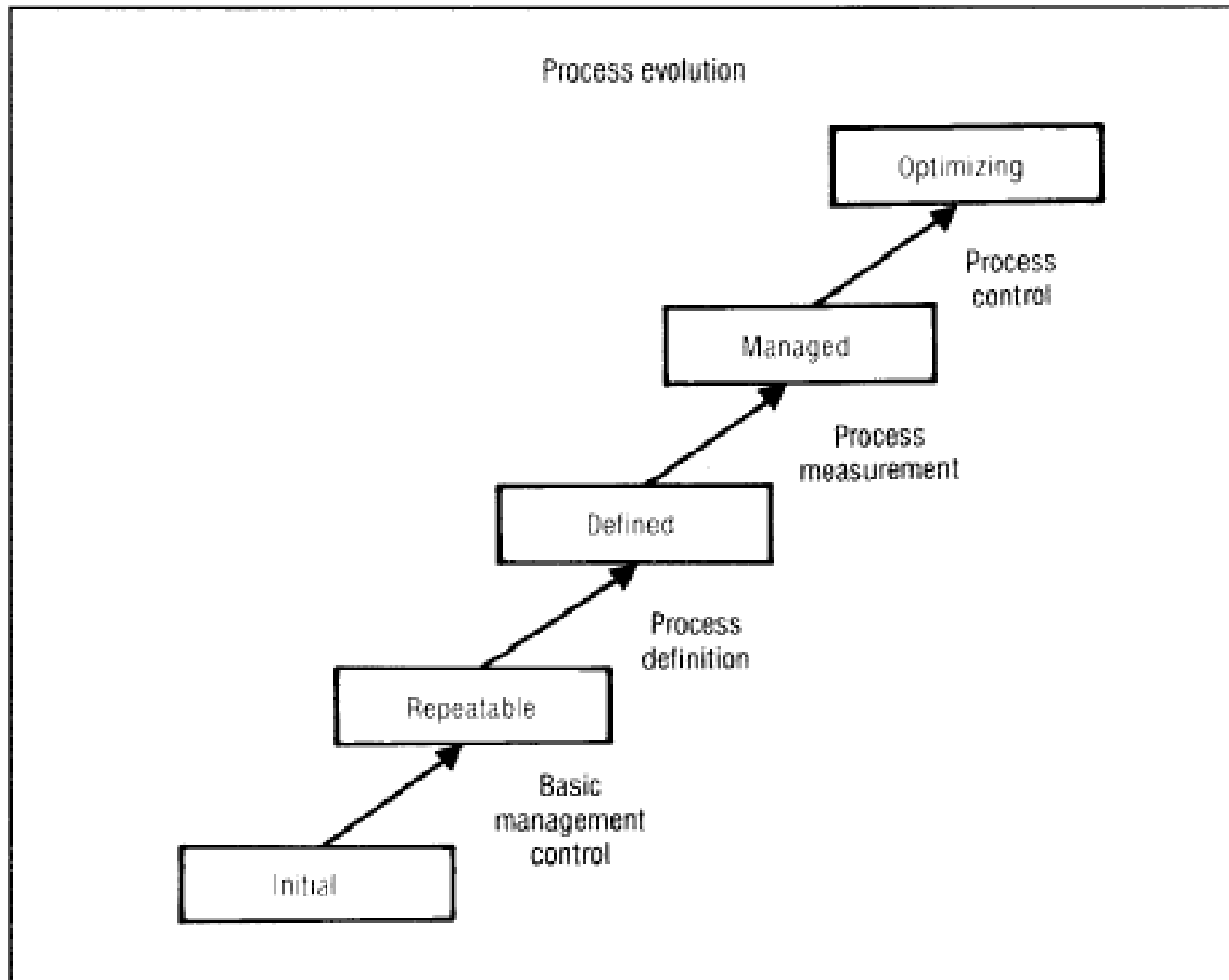
# To improve software capabilities:



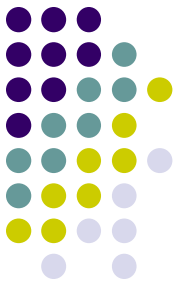
- 1. understand the current status of their development process or processes,
- 2. develop a vision of the desired process,
- 3. establish a list of required process improvement actions in order of priority,
- 4. produce a plan to accomplish these actions,
- 5. commit the resources to execute the plan.



# Process maturity levels

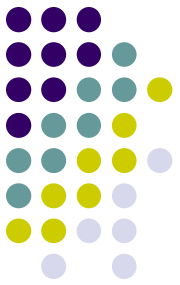


**Figure 1.** The five levels of process maturity.

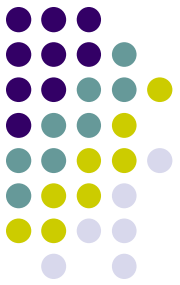


# Process maturity levels

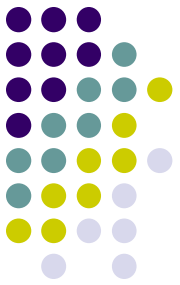
- 1. Initial. Until the process is under statistical control, no orderly progress in process improvement is possible.
- 2. Repeatable. The organization has achieved a stable process with a repeatable level of statistical control by initiating rigorous project management of commitments, cost, schedule, and changes.



- 3. Defined. The organization has defined the process, to ensure consistent implementation and provide a basis for better understanding of the process. At this point, advanced technology can usefully be introduced.
- 4. Managed. The organization has initiated comprehensive process measurements, beyond those of cost and schedule performance. This is when the most significant quality improvements begins.

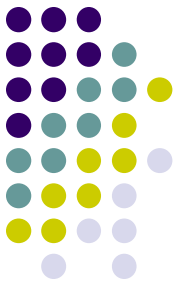


- 5. Optimizing. The organization now has a foundation for continued improvement and optimization of the process.



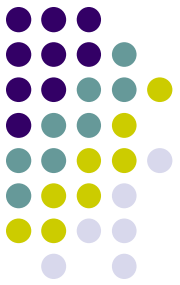
# Why select the five?

- Reasonably represent the actual historical phases of evolutionary improvement of real software organizations,
- Represent a measure of improvement that is reasonable to achieve from the prior level,
- Suggest interim improvement goals and progress measures, and
- Make obvious a set of immediate improvement priorities, once an organization's status in this framework is known.



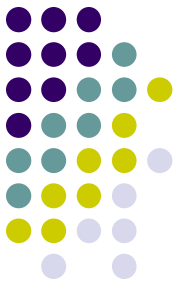
# Initial Process

- The initial Process could properly be called ad hoc, and it is often even chaotic.
- The organization typically operates without formalized procedures, cost estimation, and project plans.
- Tools are neither well integrated with the process nor uniformly applied.
- Change control is not strict and there is little senior management exposure to or understanding of the problems and issues.
- Since problems are often deferred or even forgotten, software installation and maintenance often present serious problems.

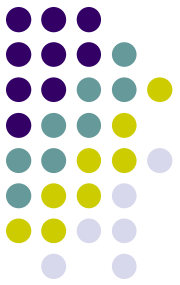


- While organizations at this level may have formal procedures for project control, there is no management mechanism to ensure they are used.
- The best test is to observe how such an organization behaves in a crisis. If it abandons established procedures and reverts to merely coding and testing, it is likely to be at the initial process level.
  - If the techniques and methods are appropriate, they must be used in a crisis and if they are not appropriate, they should not be used at all.



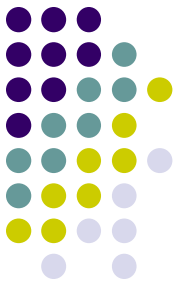


- One reason organizations behave chaotically is that they have not gained sufficient experience to understand the consequences of such behavior.
- Because many effective software actions such as design and code reviews or test data analysis do not appear to directly support shipping the product.

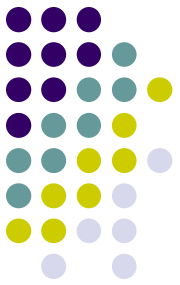


# Improve Initial Process

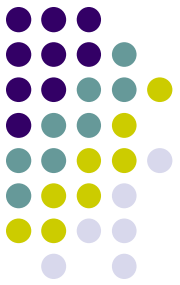
- Project management.
  - The fundamental role of a project-management system is to ensure effective control of commitments.
  - This requires adequate preparation, clear responsibility, a public declaration.



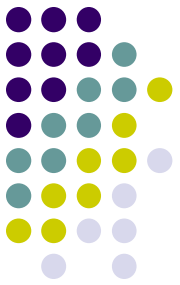
- Management oversight.
  - A disciplined software-development organization must have senior management oversight. This includes review and approval of all major development plans before official commitment.



- Quality assurance.
  - A quality-assurance group is charged with assuring management that the software-development work is actually done the way it is supposed to be done.
  - To be effective, the assurance organization must have an independent reporting line to senior management and sufficient resources to monitor performance of all key planning, implementation, and verification activities.
  - This generally requires an organization of about 5 to 6 percent the size of development organization.

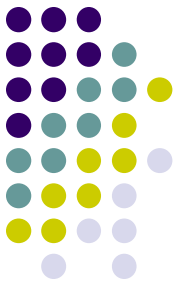


- Change control.
  - Control of changes in software development is fundamental to business and financial control as well as to technical stability.
  - To develop quality software on a predictable schedule, the requirements must be established and maintained with reasonable stability throughout the development cycle.
  - Changes will have to be made, but they must be managed and introduced in an orderly way.



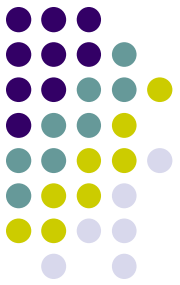
# Repeatable Process

- The Repeatable Process has one important strength over the Initial Process: It provides commitment control.
- This is such an enormous advance over the Initial Process that the people in the organization tend to believe they have mastered the software problem.
- Organizations at the Repeatable Process level face major risks when they are presented with new challenges.



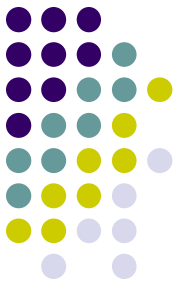
# Challenges in this level

- New tools and methods will likely affect how the process is performed, thus destroying the relevance of the intuitive historical base on which the organization relies.
- When the organization must develop a new kind of product, it is entering new territory.
  - For example, a software group that has experience developing compilers will likely have design, scheduling, and estimation problems if assigned to write a control program.



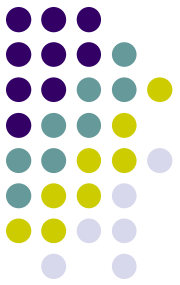
- Major organization changes can be highly disruptive. In the Repeatable Process organization, a new manager has no orderly basis for understanding what is going on and new team members must learn the ropes through word of mouth.



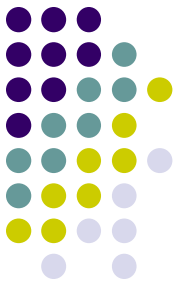


# The key to improve

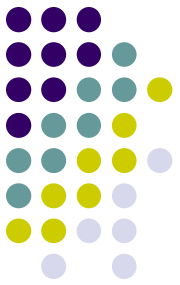
- 1. Establish a process group.
  - A process group is a technical group that focuses exclusively on improving the software-development process.
  - The responsibilities of process groups include defining the development process, identifying technology needs and opportunities, and conducting quarterly management reviews of process status and performance.



- 2. Establish a software-development process architecture that describes the technical and management activities required for proper execution of the development process.
  - The architecture is a structural decomposition of the development cycle into tasks.
  - The decomposition continues until each defined task is performed by an individual or single management unit.

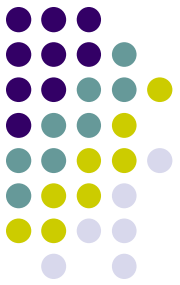


- 3. If they are not already in place, introduce a family of software-engineering methods and technologies.
  - These include design and code inspections, formal design methods, library-control systems, and comprehensive testing methods.
  - Prototyping should also be considered, along with the adoption of modern implementation languages.



# Defined Process

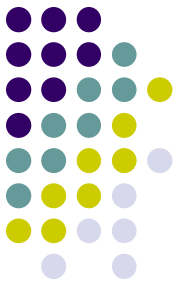
- With the Defined Process, the organization has achieved the foundation for major and continuing progress.
  - For example, the development group, when faced with a crisis, will likely continue to use the Defined Process.



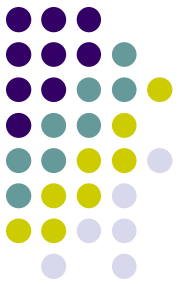
# What to improve

- Defined Process is still only qualitative: There is little data to indicate what is going on or how effective the process really is.
  - There is considerable debate about the value of software-process measurements and the best ones to use.

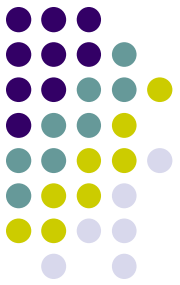
# The key steps to Managed Process.



- 1. Establish a minimum, basic set of process measurement to identify the quality and cost parameters of each process step.
  - The objective is to quantify the relative costs and benefits of each major process activity, such as the cost and yield of error detection and correction methods.

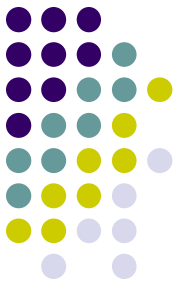


- 2. Establish a process database with the resources to manage and maintain it.
  - Cost and yield data should be maintained centrally to guard against loss, and to facilitate process quality and productivity analysis.

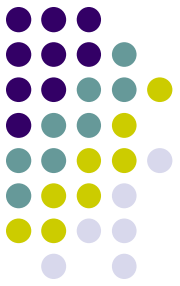


- 3. Provide sufficient process resources to gather and maintain this data and to advise project members on its use.
  - Assign skilled professionals to monitor the quality of the data before entry in the database and to provide guidance on analysis methods and interpretation.



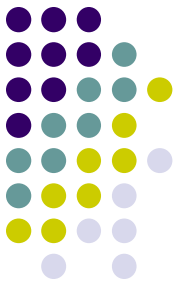


- 4. Assess the relative quality of each product and inform management where quality targets are not being met.
  - An independent quality-assurance group should assess the quality actions of each project and track its progress against its quality plan.
  - When this progress is compared with the historical experience on similar projects, and informal assessment generally can be made.

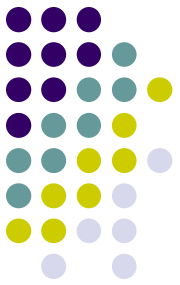


# Managed Process

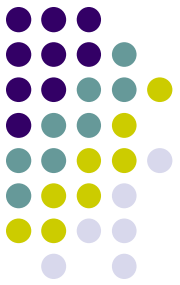
- The greatest potential problem with the Managed Process is the cost of gathering data. The data is expensive to gather and maintain.



- Approach data gathering with care and precisely define each piece of data in advance.
- Productivity data is generally meaningless unless explicitly defined.
  - New code, or changed code
  - Noncomment, nonblank lines, executable instructions, or equivalent assembler instructions.
  - How to measure management, test, documentation, and support personnel.

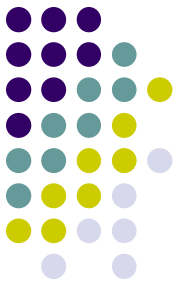


- The tendency with measured data is to use it to compare several groups and put pressure on those with the lowest ranking. This is a misapplication of process data.
- First, it is rare that two projects are comparable by any simple measures. The variations in task complexity caused by different product types can exceed five to one.

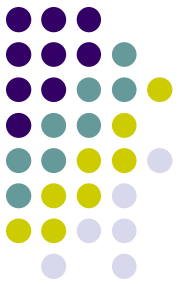


- Process data must not be used to compare projects or individuals. Its purpose is to illuminate the product being developed and to provide an informed basis for improving the process.
  - When such data is used by management to evaluate individuals or teams, the reliability of the data itself will deteriorate.
  - Few people can be counted on to provide reliable data on their own performance.

# From Managed Process to Optimizing Process

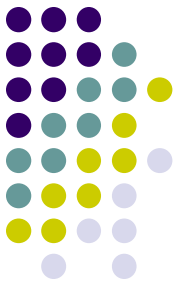


- 1. Support automatic gathering of process data. Some data cannot be gathered by hand, and all manually gathered data is subject to error and omission.
- 2. Use this data to both analyze and modify the process to prevent problems and improve efficiency.



# Optimizing Process

- Up to this point, software-development managers have largely focused on their products and will typically only gather and analyze data that directly relates to product improvement.
- In the Optimizing Process, the data is available to actually tune the process itself.

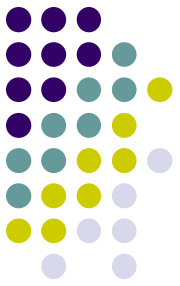


# Fixing bugs example

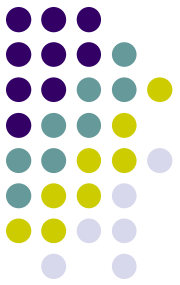
- Author's experience: it takes about one to four working hours to find and fix a bug through inspections and about 15 to 20 working hours to find and fix a bug in function or system test.
- However, some kinds of errors are either uneconomical or almost impossible to find except by machine.
  - Spelling, interfaces, performance, human factors...



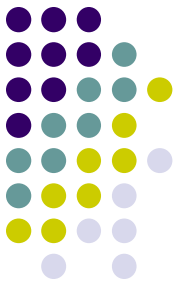
# Optimizing Process gives us a new perspective on testing



- To reduce the cost of removal of bugs, inspections should be emphasized together with any other cost-effective techniques.
- The role of functional and system testing should then be changed to one of finding symptoms that are further explored to see if the bug is an isolated problem or if it indicates design problems that require more comprehensive analysis.

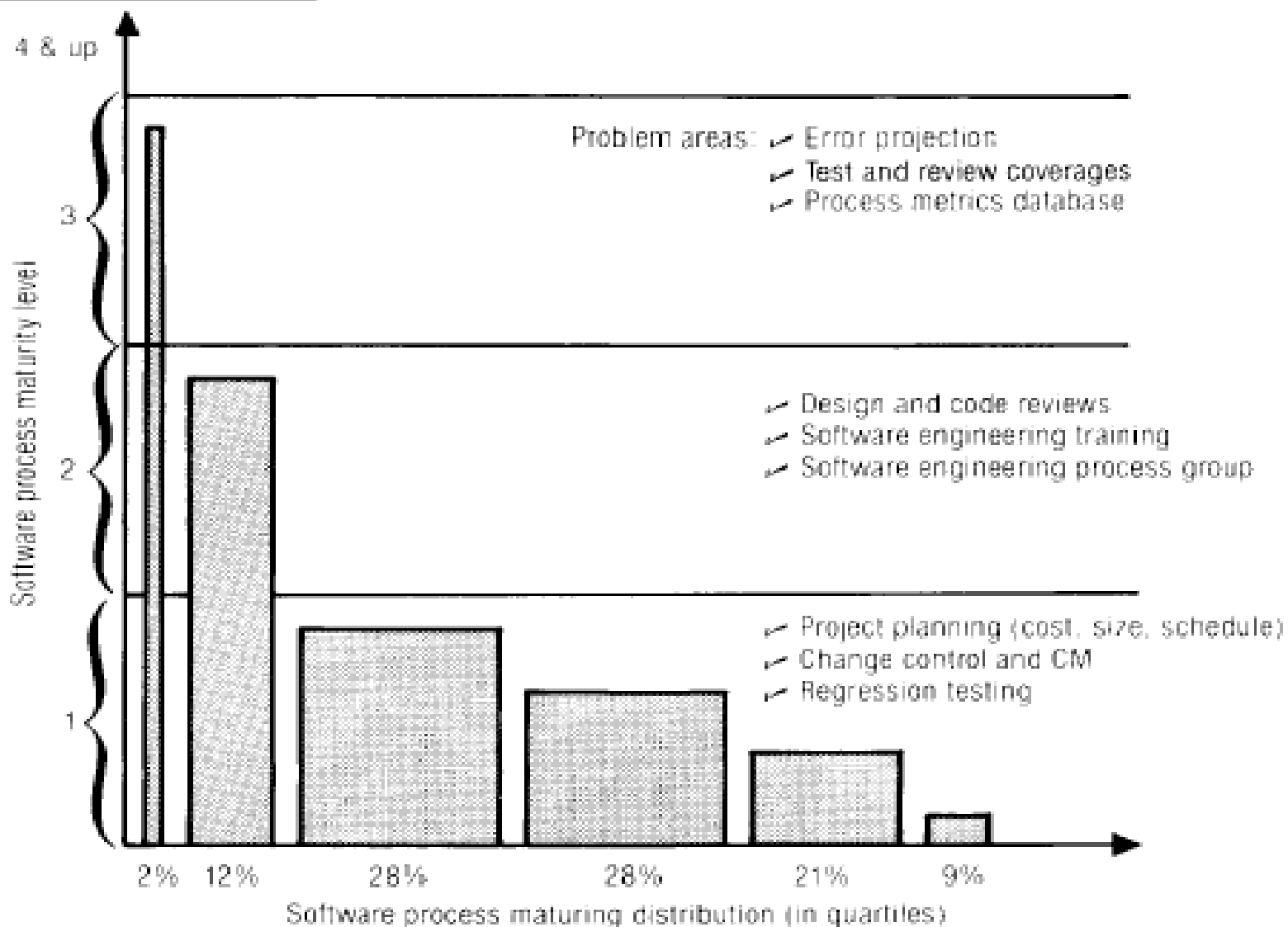
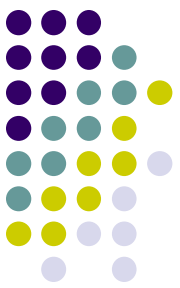


- In the Optimizing Process, the organization has the means to identify the weakest elements of the process and fix them.
- At this point, data is available to justify the application of technology to various critical tasks and numerical evidence is available on the effectiveness with which the process has been applied to any given product.



# People in the process

- The Optimizing Process helps in several ways:
  - It helps managers understand where help is needed and how best to provide the people with the support they require.
  - It lets professionals communicate in concise, quantitative terms.
  - It provides the frame work for the professionals to understand their work performance and to see how to improve it.



**Figure 2.** Early results from several dozen software organizations queried by the SEI shows the maturity distribution and the three leading problems faced at each level. At level one, the distribution is shown by quartile. There is not yet sufficient data to provide this detail for levels 2 or 3. To date, no complete organizations have been observed at levels 4 or 5.