Chapter 3

Software Process Improvement

Contents

- What Is SPI?
  - Approaches for SPI, Maturity Models, Is SPI for Everyone, Benefits of SPI
- The SPI Process
  - Assessment and Gap Analysis, Education and Training, Selection and Justification, Installation/Migration, Evaluation
  - Risk Management for SPI
- The CMM
  - Why CMM, Five Levels of CMM
  - Internal Structure to Maturity Levels
- Other SPI Frameworks
- SPI Return on Investment
- SPI Future Trends

What is SPI?

- The quality of the software system is governed by the quality of the process used to develop and maintain it.
  - Good SE => Good Development -> Good product
- Software Process Improvement (SPI) involves
  - understanding existing processes and
  - introducing process changes
    - to improve product quality,
    - reduce costs or
    - accelerate schedules
- The SPI strategy transforms the existing approach to software development into something that is **more focused**, **more repeatable**, and **more reliable**

What is SPI?...

- Most process improvement work so far has focused on defect reduction.
    - This reflects the increasing attention paid by industry to quality.
- However, other process attributes can also be the focus of improvement
- **These attributes are**
    - **Understandability** - To what extent is the process explicitly defined and how easy is it to understand the process definition?
    - **Visibility** - Do the process activities culminate in clear results so that the progress of the process is externally visible?
    - **Supportability** - To what extent can CASE tools be used to support the process activities?

What is SPI?...

- **Acceptability** - Is the defined process acceptable to and usable by the engineers responsible for producing the software product?
- **Reliability** - Is the process designed in such a way that process errors are avoided or trapped before they result in product errors?
- **Robustness** - Can the process continue in spite of unexpected problems?
- **Maintainability** - Can the process evolve to reflect changing organizational requirements or identified process improvements?
- **Rapidity** - How fast can the process of delivering a system from a given specification be completed?

What is SPI?...

- SPI implies that
  - elements of an effective software process can be defined in an effective manner
  - an existing organizational approach to software development can be assessed against those elements, and
  - a meaningful strategy for improvement can be defined.
- The effort and time that is required to implement a SPI strategy must pay for itself in some measurable way.
- That is, SPI Strategy must reduce
  - the **number of defects** that are delivered to end users,
  - the **amount of rework** due to quality problems,
  - the **costs** associated with software maintenance and support
  - the **indirect costs** that occur when software is delivered late.

Approaches for SPI

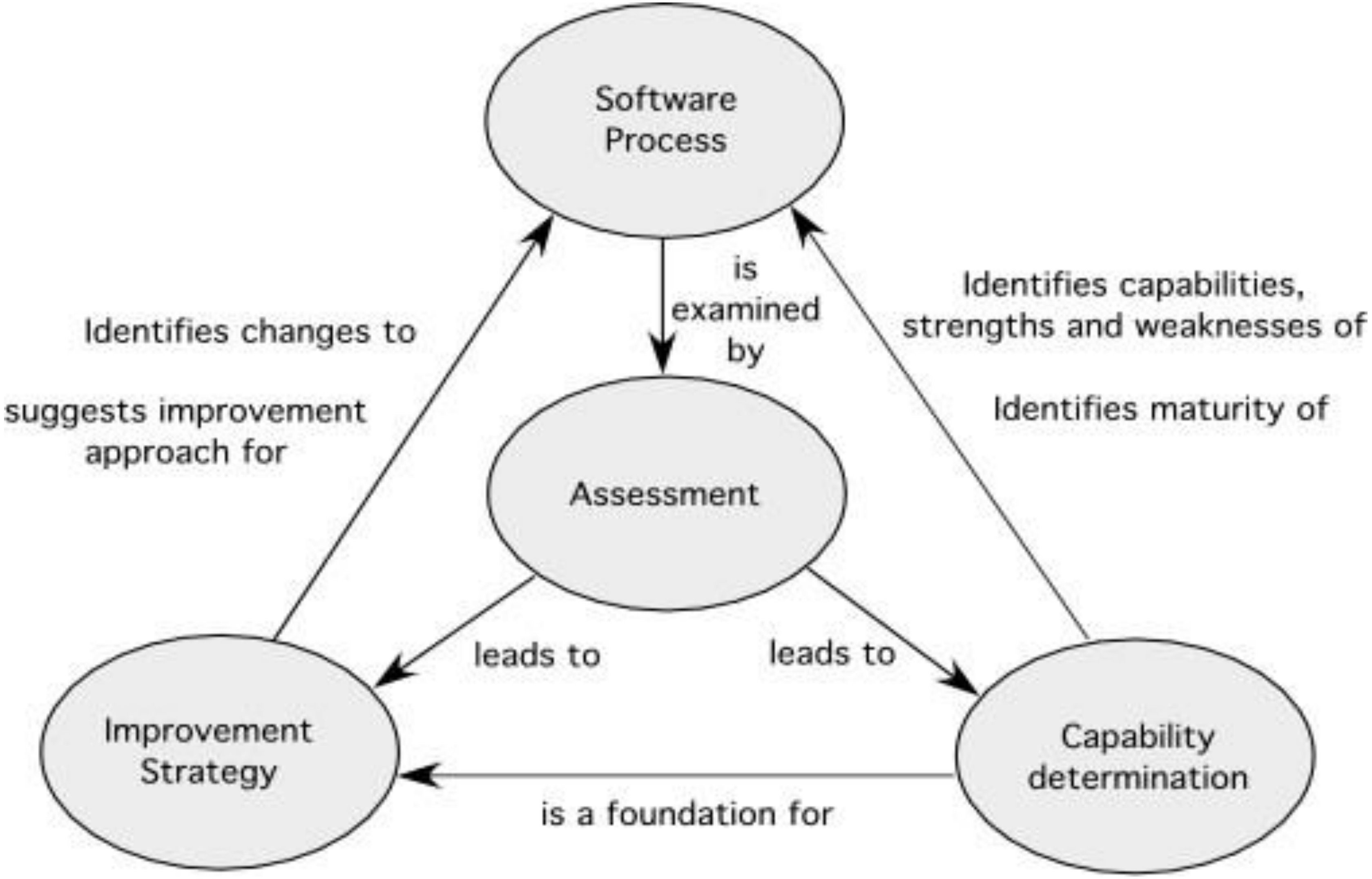
- The process maturity approach, which focuses on improving process and project management and introducing good software engineering practice.
  - The level of process maturity reflects the extent to which good technical and management practice has been adopted in organizational software development processes.
    - The primary goal of this approach are improved product quality and process predictability
- The agile approach, which focuses on iterative development and the reduction of overheads in the software process.
  - The primary characteristics of agile methods are rapid delivery of functionality and responsiveness to changing customer requirements.

Approaches for SPI

- Informal approach to SPI can be chosen, but the vast majority choose one of a number of SPI frameworks
- An *SPI framework* defines
  - a **set of characteristics** that must be present if an effective software process is to be achieved
  - a **method for assessing** whether those characteristics are present
  - a **mechanism for summarizing the results** of any assessment, and
  - a **strategy for assisting** a software organization in implementing those process characteristics that have been found to be weak or missing.
- SPI framework assesses the “maturity” of an organization’s software process and provides a qualitative indication of a maturity level

Approaches for SPI…

Elements of a SPI Framework



Maturity Models

- A **maturity model** is applied within the context of an SPI framework.
- The intent of the maturity model is to provide an overall indication of the "process maturity" exhibited by a software organization.
  - an indication of the quality of the software process, the degree to which practitioner's understand and apply the process, and the general state of software engineering practice.
- This is accomplished using some type of ordinal scale.
  - SW-CMM and ISO9001
- Other standards help with process assessment, capability determination, and process improvement are
  - SPICE, ISO/IEC 15504 and ISO/IEC 12207

Is SPI for Everyone?

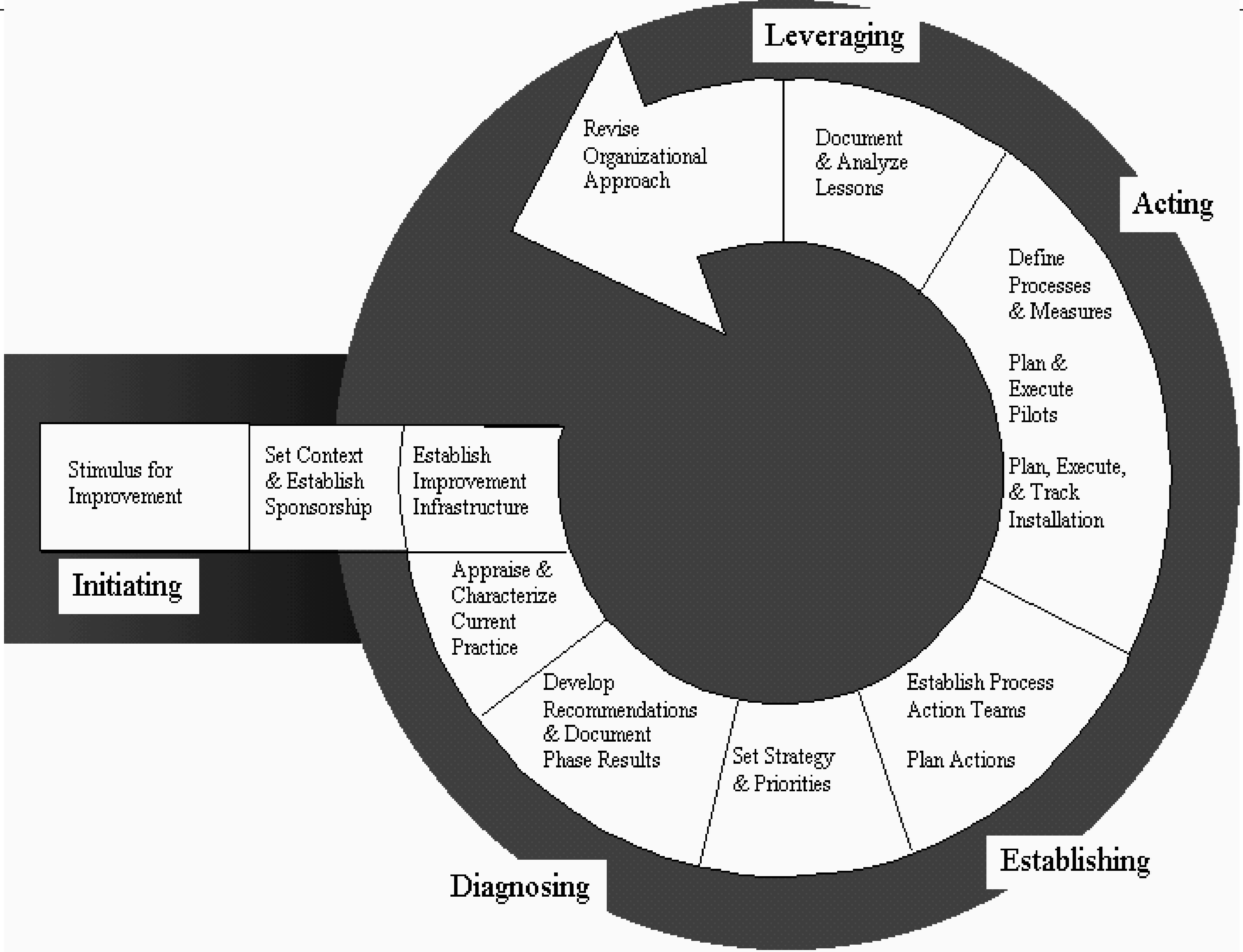
- For many years, SPI was viewed as a “corporate” activity
- But, can a small company initiate SPI activities and do it successfully?
  - Answer: a qualified “yes”
- It should come as no surprise that small organizations are more informal, apply fewer standard practices, and tend to be self-organizing.
  - SPI will be approved and implemented only after its proponents demonstrate *financial leverage.*
  - That is you must show a realistic return on investment for SPI costs.

Benefits of SPI ?

- Benefits of SPI
  - Improvements to quality
  - Reductions in the cost of poor quality
  - Improvements in productivity
  - Reductions to the cost of software development
  - Improvements to on-time delivery
  - Improved consistency in budget and schedule delivery
  - Improvements to customer satisfaction
  - Improvements to employee morale

SPI Process

- Establishing a consensus for initiating SPI and defining an ongoing strategy for implementing it across a software organization is a very difficult process.
- To alleviate the problem SEI has defined SPI Implementation Model called **IDEAL**
  - It is an organizational improvement model that serves as a roadmap for initiating, planning, and implementing improvement actions
  - Depicts activities of a process improvement programme and presents a consistent view of what is involved in transitioning the CMM into an organization's practice.
  - The five distinct activities that guide an organization through SPI activities are **I**nitiating, **D**iagnosing, **E**stablishing, **A**cting, and **L**earning



Initiating

Stimulus for Improvement

Set Context & Establish Sponsorship

Establish Improvement Infrastructure

Diagnosing

Appraise & Characterize Current Practice

Develop Recommendations & Document Phase Results

Establishing

Set Strategy & Priorities

Establish Process Action Teams

Plan Actions

Acting

Define Processes & Measures

Plan & Execute Pilots

Plan, Execute, & Track Installation

Leveraging

Document & Analyze Lessons

Revise Organizational Approach

SPI Process...

- In this lecture we will see in detail somewhat different road map for SPI
- It applies a commonsense philosophy that requires an organization to
  1. look in the mirror,
  2. then get smarter so it can make intelligent choices,
  3. select the process model (and related technology elements) that best meets its needs,
  4. instantiate the model into its operating environment and its culture, and
  5. evaluate what has been done.
- These five activities are applied in an iterative (cyclical) manner in an effort to foster continuous process improvement

SPI Process: Assessment and Gap Analysis

- Assessment examines a wide range of actions and tasks that will lead to a high quality process.
- **Consistency.** Are important activities, actions and tasks applied consistently across all software projects and by all software teams?
- **Sophistication.** Are management and technical actions performed with a level of sophistication that implies a thorough understanding of best practice?
- **Acceptance.** Is the software process and software engineering practice widely accepted by management and technical staff?
- **Commitment.** Has management committed the resources required to achieve consistency, sophistication and acceptance?
- Gap analysis—The difference between local application and best practice represents a "gap" that offers opportunities for improvement.

SPI Process: Education and Training

- Three types of education and training should be conducted:
    - **Generic concepts and methods.** Directed toward both managers and practitioners, this category stresses both process and practice. The intent is to provide professionals with the intellectual tools they need to apply the software process effectively and to make rational decisions about improvements to the process.
    - **Specific technology and tools.** Directed primarily toward practitioners, this category stresses technologies and tools that have been adopted for local use. For example, if UML has been chosen for analysis and design modeling, a training curriculum for software engineering using UML would be established.
    - **Business communication and quality-related topics.** Directed toward all stakeholders, this category focuses on “soft” topics that help enable better communication among stakeholders and foster a greater quality focus.

SPI Process: Selection and Justification

- Choose the process model that best fits your organization, its stakeholders, and the software that you build
- Decide on the set of framework activities that will be applied, the major work products that will be produced and the quality assurance checkpoints that will enable your team to assess progress
- Develop a work breakdown for each framework activity (e.g., modeling), defining the task set that would be applied for a typical project
- Once a choice is made, time and money must be expended to install it within an organization and these resource expenditures should be justified.

SPI Process: Installation/Migration

- *Installation* - Framework activities, software engineering actions, and individual work tasks must be defined and installed as part of a new software engineering culture.
- *Migration* - changes associated with SPI are relatively minor, representing small, but meaningful modifications to an existing process model
- *Software process redesign* (SPR) - concerned with identification, application, and refinement of new ways to dramatically improve and transform software processes.
- Three different process models are considered:
  - the existing ("as-is") process,
  - a transitional ("here-to-there") process, and
  - the target ("to be") process.

SPI Process: Evaluation

- Assesses
  - the degree to which changes have been instantiated and adopted,
  - the degree to which such changes result in better software quality or other tangible process benefits, and
  - the overall status of the process and the organizational culture as SPI activities proceed
- From a qualitative point of view, past management and practitioner attitudes about the software process can be compared to attitudes polled after installation of process changes.

Risk Management for SPI

- Manage risk at three key points in the SPI process
  - prior to the initiation of the SPI roadmap,
  - during the execution of SPI activities (assessment, education, selection, installation), and
  - during the evaluation activity that follows the instantiation of some process characteristic.
- In general, the following categories can be identified for SPI risk factors:
  - budget and cost
  - content and deliverables culture
  - maintenance of SPI deliverables
  - mission and goals
  - organizational management and
  - organizational stability
  - process stakeholders
  - schedule for SPI development
  - SPI development environment and process
  - SPI project management and SPI staff

Risk Critical Success Factors

- SPI is a risky endeavor and that the failure rate for companies that try to improve their process is distressingly high.
- Organizational risks, people risks, and project management risks are present challenges for those who lead any SPI effort.
- Although risk management is important, it’s equally important to recognize those critical factors that lead to success.
- The top five Critical Success Factors are
  - Management commitment and support
  - Staff involvement
  - Process integration and understanding
  - A customized SPI strategy
  - Solid management of the SPI project

“If you don’t know where you are, a map won’t help”

Watts Humprey

Capability Maturity Model(CMM)

- Developed by the Software Engineering Institute (SEI) of the Carnegie Mellon University
- Framework that describes the key elements of an effective software process.
- Describes an evolutionary improvement path for software organizations from an ad hoc, immature process to a mature, disciplined one.
- Provides guidance on how to gain control of processes for developing and maintaining software and how to evolve toward a culture of software engineering and management excellence.

CMM…

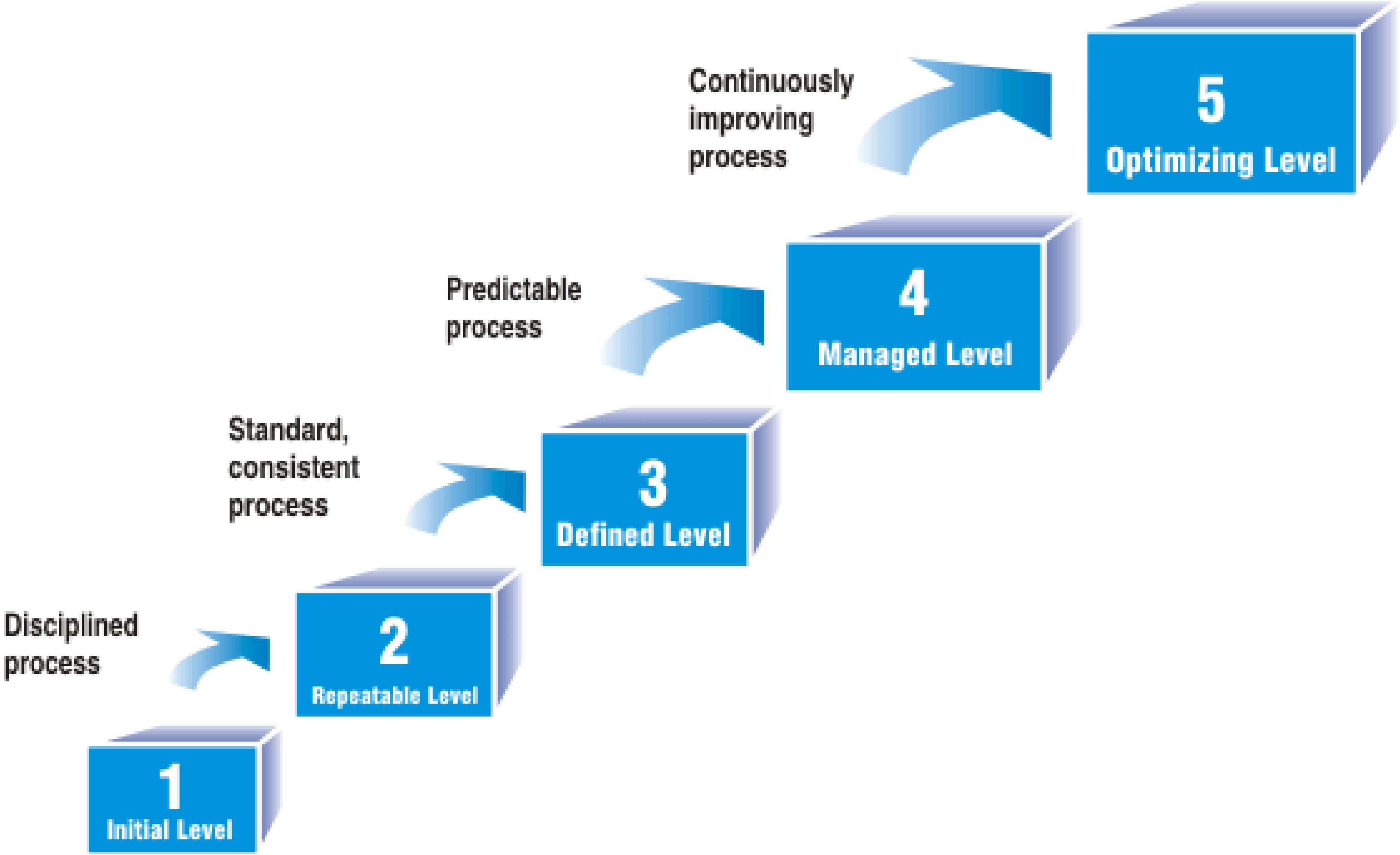
- **Matured Organizations**
  - Processes are defined, documented and controlled
  - Roles and responsibilities are clear
  - Products and processes are measured
  - Quality, costs and schedules are measured and followed-up
  - Management is committed to continuous improvement
  - Technology is effectively used within organisation’s SW process(es)
  - Preventive quality work is a fact

CMM...

- **Why CMM?**
  - Increasing cost of software
  - Quality problems in software products
  - Cost of software maintenance
  - Governments put billions of dollars in software acquisition
  - Countries competitiveness increasingly dependent on software
  - Increasing rate of change in technology and software environment
  - Typical software project was a year late and exceeded two times the budget
  - Increasing SW complexity

CMM...

- Five Levels of CMM



CMM Level 1: Initial

- The software process is characterized as ad hoc, and occasionally even chaotic.
- Few processes are defined, and success depends on **individual effort.**
    - At this level, frequently have difficulty making commitments that the staff can meet with an orderly process
    - Products developed are often over budget and schedule
    - Wide variations in cost, schedule, functionality and quality targets
    - Capability is a characteristic of the individuals, not of the organization

CMM Level 2: Repeatable

- Basic process management processes are established to track cost, schedule, and functionality.
- The necessary process discipline is in place to repeat earlier successes on projects with similar applications.
  - Realistic project commitments based on results observed on previous projects
  - Software project standards are defined and faithfully followed
  - Processes may differ between projects
  - Process is disciplined
  - earlier successes can be repeated

CMM Level 3: Defined

- The software process for both management and engineering activities is
  - documented,
  - standardized, and
  - integrated
  - into a standard software process for the organization.
- All projects use an approved, tailored version of the organization's standard software process for developing an maintaining software.
- Project and organisation levels training plans are created and followed
- More systematic technical coordination between different project groups

CMM Level 4: Managed

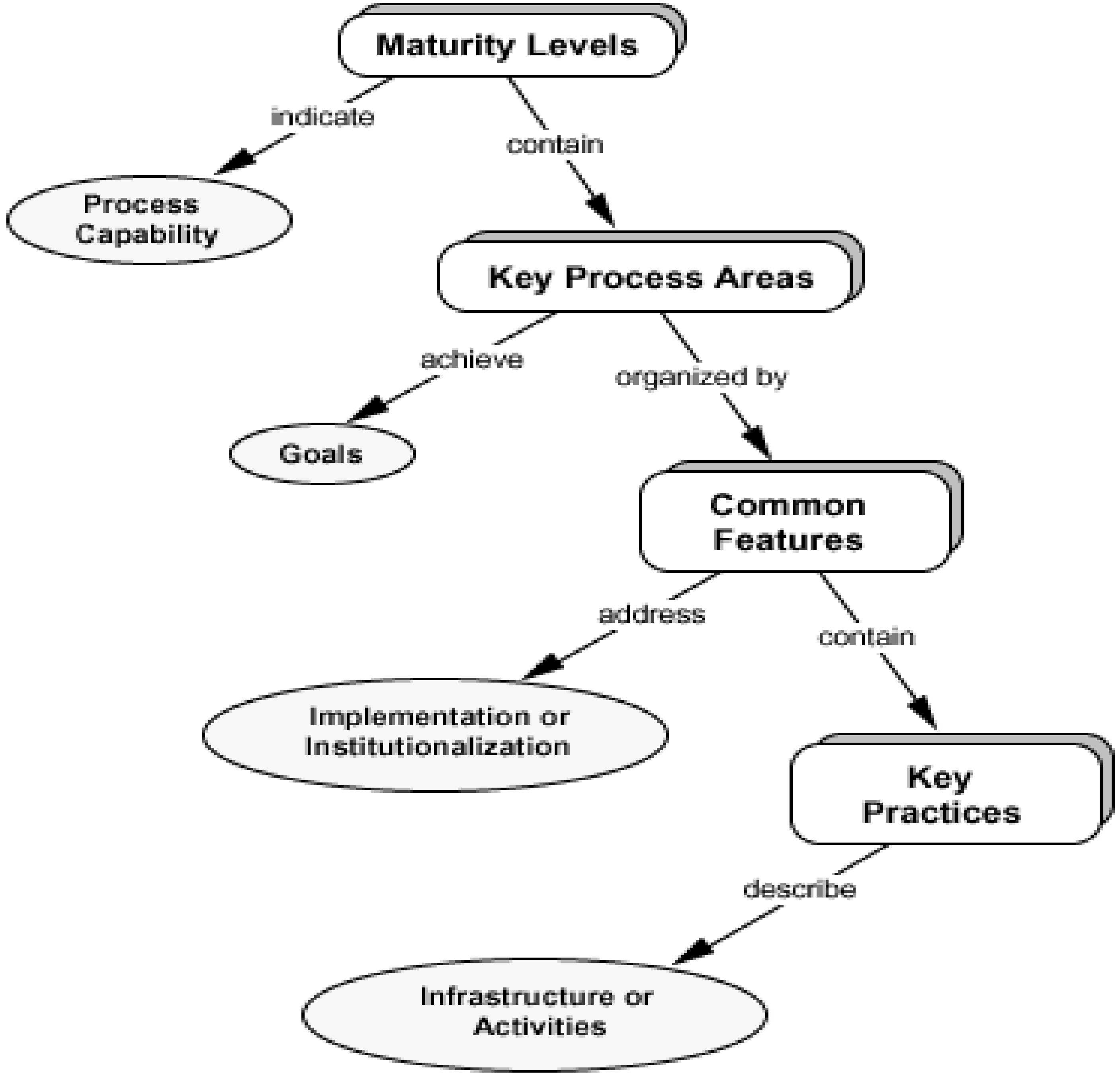
- Detailed measures of the software process and product quality are collected.
- Both the software process and products are quantitatively understood and controlled.
  - Narrowing the variation in process performance to fall within acceptable quantitative bounds
  - When known limits are exceeded, corrective action can be taken
  - Quantifiable and predictable
    - predict trends in process and product quality

CMM Level 5: Optimizing

- Continuous process improvement is enabled by quantitative feedback from the process and from piloting innovative ideas and technologies.
- Goal is to prevent the occurrence of defects
- Causal analysis
- Data on process effectiveness used for cost benefit analysis of new technologies and proposed process changes
- The causes of defects are eliminated as part of preventive quality work
- New technologies can be utilized effectively to improve process capability

CMMI model components

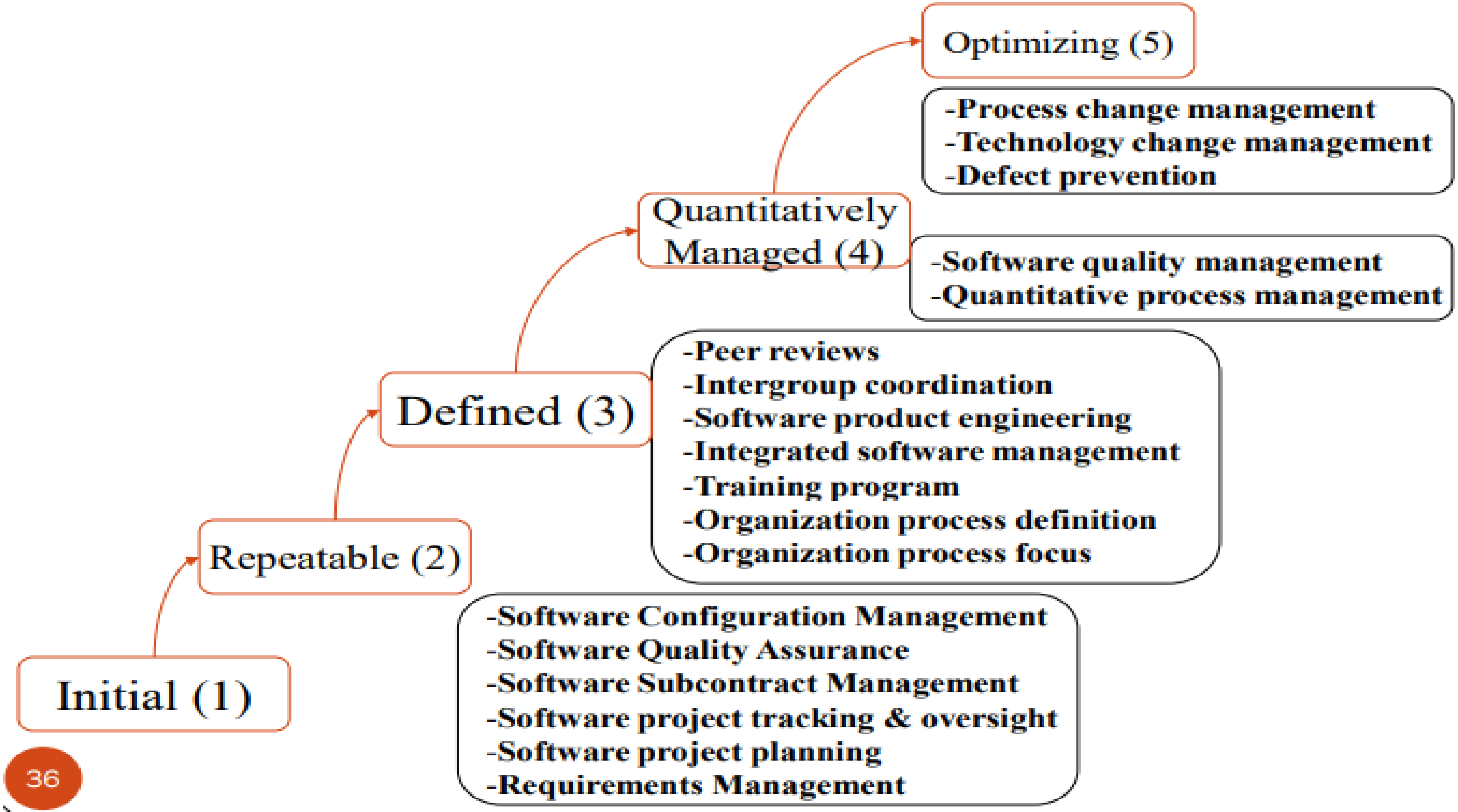
- Process areas
  - process areas that are relevant to process capability and improvement are identified.
- Goals
  - Goals are descriptions of desirable organizational states. Each process area has associated goals.
- Practices
  - Practices are ways of achieving a goal - however, they are advisory and other approaches to achieve the goal may be used


Maturity Levels
indicate
contain
Process Capability
Key Process Areas
achieve
organized by
Goals
Common Features
address
contain
Implementation or Institutionalization
Key Practices
describe
Infrastructure or Activities

Key Process Areas - KPAs

- Except for level 1, each level is decomposed into key process areas (KPA)
- Key process areas(KPAs) identify areas on which an organization should focus in order to improve its software development processes.
- Each KPA identifies a cluster of related activities that, when performed collectively, achieve a set of goals considered important for enhancing software capability.

Key Process Areas...



Optimizing (5)

- -Process change management
- -Technology change management
- -Defect prevention

Quantitatively Managed (4)

- -Software quality management
- -Quantitative process management

Defined (3)

- -Peer reviews
- -Intergroup coordination
- -Software product engineering
- -Integrated software management
- -Training program
- -Organization process definition
- -Organization process focus

Repeatable (2)

- -Software Configuration Management
- -Software Quality Assurance
- -Software Subcontract Management
- -Software project tracking & oversight
- -Software project planning
- -Requirements Management

Initial (1)

Level 2 KPAs

- Requirements Management
  - Establish common understanding of customer requirements between the customer and the software project
  - Requirements is basis for planning and managing the software project
  - Not working backwards from a given release date!
- Software Project Planning
  - Establish reasonable plans for performing the software engineering activities and for managing the software project
- Software Project Tracking and Oversight
  - Establish adequate visibility into actual progress
  - Take effective actions when project’s performance deviates significantly from planned

Level 2 KPAs...

- Software Subcontract Management
  - Manage projects outsourced to subcontractors
- Software Quality Assurance
  - Provide management with appropriate visibility into
    - process being used by the software projects
    - work products
- Software Configuration Management
  - Establish and maintain the integrity of work products
  - Product baseline
  - Baseline authority

Level 3 KPAs

- Organization Process Focus
  - Establish organizational responsibility for software process activities that improve the organization’s overall software process capability
- Organization Process Definition
  - Develop and maintain a usable set of software process assets
    - stable foundation that can be institutionalized
    - basis for defining meaningful data for quantitative process management
- Training Program
  - Develop skills and knowledge so that individual can perform their roles effectively and efficiently
  - Organizational responsibility
  - Needs identified by project

Level 3 KPAs...

- Integrated Software Management
  - Integrated engineering and management activities
  - Engineering and management processes are tailored from the organizational standard processes
  - Tailoring based on business environment and project needs
- Software Product Engineering
  - technical activities of the project are well defined (SDLC)
  - correct, consistent work products
- Intergroup Coordination
  - Software engineering groups participate actively with other groups
- Peer Reviews
  - early defect detection and removal
  - better understanding of the products
  - implemented with inspections, walkthroughs, etc

Level 4 KPAs

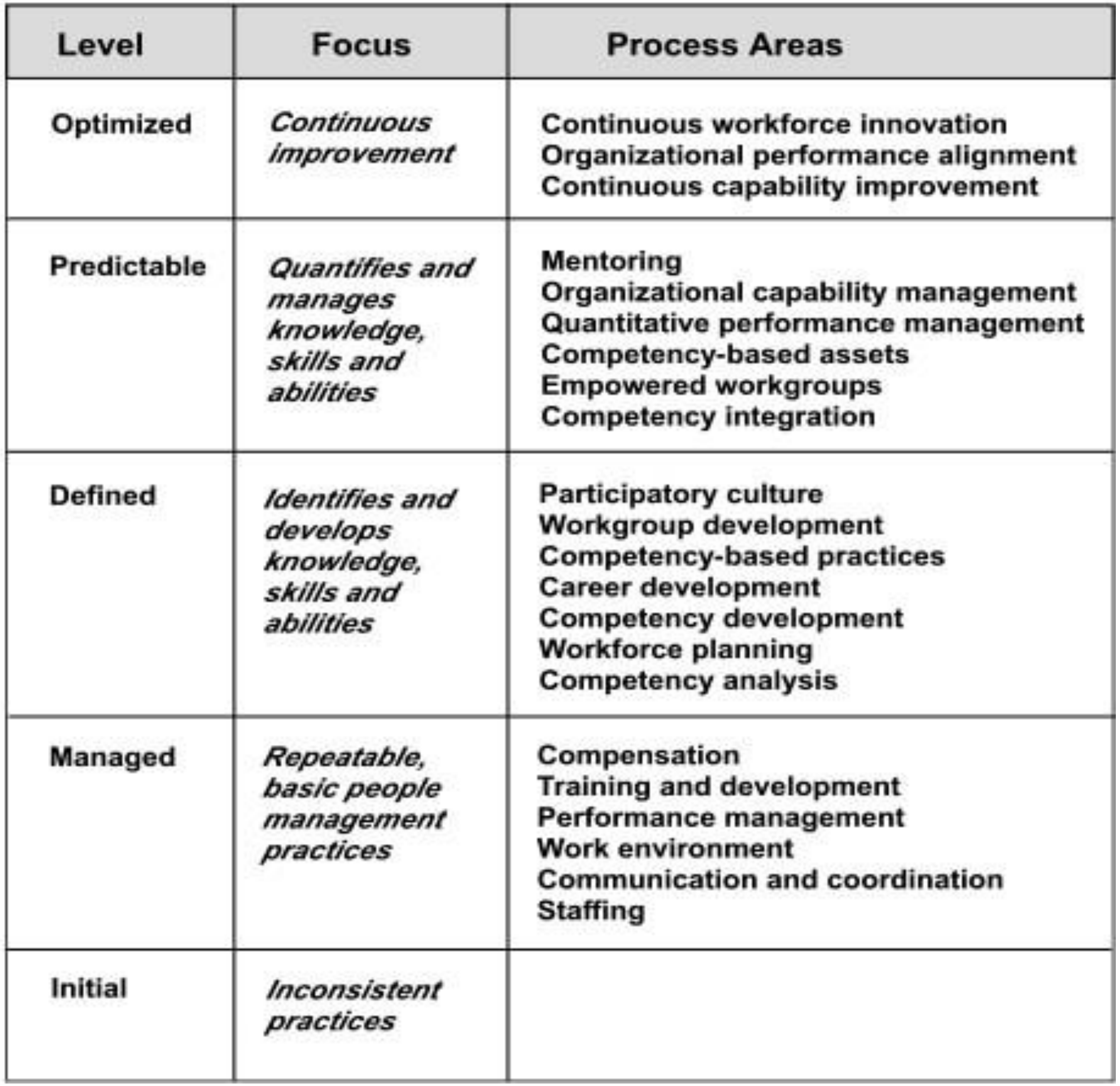
- Quantitative Process Management
  - control process performance quantitatively
  - actual results from following a software process
  - focus on identifying and correcting special causes of variation with respect to a baseline process
- Software Quality Management
  - quantitative understanding of software quality
    - products
    - process

Level 5 KPAs

- Process Change Management
  - continuous process improvement to improve quality, increase productivity, decrease cycle time
- Technology Change Management
  - identify and transfer beneficial new technologies
    - tools
    - methods
    - processes
- Defect Prevention
  - causal analysis of defects to prevent recurrence

The People CMM

- is a maturity framework that focuses on continuously improving the management and development of the human assets of an organization
- defines a set of five organizational maturity levels that provide an indication of the relative sophistication of workforce practices and processes



Level	Focus	Process Areas
Optimized	*Continuous improvement*	Continuous workforce innovation Organizational performance alignment Continuous capability improvement
Predictable	*Quantifies and manages knowledge, skills and abilities*	Mentoring Organizational capability management Quantitative performance management Competency-based assets Empowered workgroups Competency integration
Defined	*Identifies and develops knowledge, skills and abilities*	Participatory culture Workgroup development Competency-based practices Career development Competency development Workforce planning Competency analysis
Managed	*Repeatable, basic people management practices*	Compensation Training and development Performance management Work environment Communication and coordination Staffing
Initial	*Inconsistent practices*	

Other SPI Frameworks

- **Capability Maturity Integration**(CMMI)
  - is a process model that provides a clear definition of what an organization should do to promote behaviors that lead to improved performance.
  - It is successor of CMM

- **Software Process Improvement and Capability determination(SPICE)**
  - an international initiative to support the International Standard ISO/IEC 15504 for (Software) Process Assessment

Other SPI Frameworks...

- **Bootstrap**
  - a SPI framework for small and medium sized organizations that conforms to SPICE
- **Team Software Process(TSP) & Personal Software Process (PSP)**
  - individual and team specific SPI frameworks that focus on process in-the-small, a more rigorous approach to software development coupled with measurement
- **TickIT**
  - an auditing method that assesses an organization compliance to ISO Standard 9001:2000

SPI Return on Investment

- “How do I know that we’ll achieve a reasonable return for the money we’re spending?”
  - ROI = [S (*benefits*) – S (*costs*)] / S (*costs*)] x 100%
- where
  - *ROI* return on Investment
  - *benefits* include the cost savings associated with higher product quality (fewer defects), less rework, reduced effort associated with changes, and the income that accrues from shorter time-to-market.
  - *costs* include both direct SPI costs (e.g., training, measurement) and indirect costs associated with greater emphasis on quality control and change management activities and more rigorous application of software engineering methods (e.g., the creation of a design model).

SPI Trends

- Future SPI frameworks must become significantly more agile
- Rather than an organizational focus (that can take years to complete successfully), contemporary SPI efforts should focus on the project level
- To achieve meaningful results (even at the project level) in a short time frame, complex framework models may give way to simpler models.
- Rather than dozens of key practices and hundreds of supplementary practices, an agile SPI framework should emphasize only a few pivotal practices