Concerns and Control of Software Engineering Practice

©Joey Paquet 2000, 2002

Overview

- Processes
- The Software Crisis
- The Capability Maturity Model (CMM)
- Drawbacks of CMM
- Conclusions

©Joey Paquet 2000, 2002

What Makes ***You*** Better?

- Why are *you* a better programmer than a first-year student?

©Joey Paquet 2000, 2002

Some Answers...

- More training
  - Know how to use tools
  - Have seen some problems before
  - Know how things fit together
- Familiar with language
  - Object, semaphore, recursion, etc
- **Better process!**

©Joey Paquet 2000, 2002

Examples of Better Processes

- Incremental coding
  - Code a little, test a little
  - Compare to writing *entire* program and testing
- Error prevention versus debugging
  - Cheaper and easier to prevent than to find bugs

©Joey Paquet 2000, 2002

What Makes Organizations Better?

- Why are some organizations better than others?
  - Deliver software on time
  - Deliver with high quality and few defects
- Do some of the same things for a personal level scale up to organizations?
- Assume two groups are doing same project, why would one be better than another?

©Joey Paquet 2000, 2002

Some Answers...

- Great people
- Smart management
- Better tools
- Good processes, standards and policies that all team members know and follow

©Joey Paquet 2000, 2002

What is the Point of Processes?

- Basically, so we don't reinvent the wheel!
- Learn from mistakes
  - Never make the same mistake twice!
- Incorporate best practices
  - Something works better than another
- Routinization of standard tasks
  - Do it right once and then reuse it
- **Predictability**

©Joey Paquet 2000, 2002

The Software Crisis

- Software is late, over budget, and low quality
- “Software Engineering” was coined in 1960s to apply engineering practices to software
- Third generation of hardware more powerful, led to larger systems
- Need new techniques and methods to control complexity

©Joey Paquet 2000, 2002

Some Current Statistics

- Programmers tend to underestimate their tasks by 20 to 30% (van Genuchten 1991)
- The average small-project estimate is off by more than 100 % (Standish Group 1994)
- The average large project is a year late (Jones 1994)
- Less than 14% of projects larger than [~12,500,000 LOC in C] delivered on time (Jones)

©Joey Paquet 2000, 2002

More Current Statistics

	%Late	%Cancelled
Small	14%	28%
Large	24%	48%
Really Large	21%	65%

From Ed Yourdon

©Joey Paquet 2000, 2002

Goals of SE

- Improve the productivity of the development process
- Improve the comprehension of the developed software systems
- Improve the quality of the software product at all levels
  - Reliability
  - Efficiency (Speed, resource usage)
  - User-friendly (user acceptance)
  - Maintainability (comprehensive design and documentation)

- **General goal:** to produce quality software which is economic and useful and safe for people.

©Joey Paquet 2000, 2002

Concerns of SE

- Products
  - Software products, test drivers (internal and external)
  - Paper documents (internal and external)
- Processes
  - How software is created (plan, tools, techniques)
  - How the quality is evaluated and ensured
- Tools
  - CASE tools, editors, project management tools, etc.
- People
  - Technical, social, and managerial skills
- Principles
  - Providing repeatability, guidelines and maturity in the software development process

©Joey Paquet 2000, 2002

SEI-CMM

- Software Engineering Institute Capability Maturity Model (SEI-CMM, or CMM)
- CMM was developed by SEI at the request of the DoD in 1987
- Based on
  - Statistical quality control (Deming's TQM, Juran)
  - Quality management (Crosby)
  - Feedback from industry and government projects

©Joey Paquet 2000, 2002

Why CMM?

- Government needs predictability
  - time, cost, quality
- Premises
  - The process of constructing software can be defined, managed, measured, and progressively improved
  - In a mature and adaptable process, people, methods, techniques, and technology are coupled to consistently produce quality software

©Joey Paquet 2000, 2002

Why CMM?

- **Mature process**
- Disciplined
- Predictable
- Proven methods
- Managed
- Metrics for
  - Quality
  - Schedule
  - Cost
  - Functionality

- **Immature process**
- Ad hoc
- Unpredictable
- Improvised efforts
- Reactionary
- Few or no metrics

©Joey Paquet 2000, 2002

CMM Organization

- Evolutionary path to increase software process maturity
  - Guide organizations in selecting improvement strategies
  - Small but continuous improvements
- Descriptive, not prescriptive
  - Describes goals but not how to achieve them
- Maturity characterized in five levels

©Joey Paquet 2000, 2002

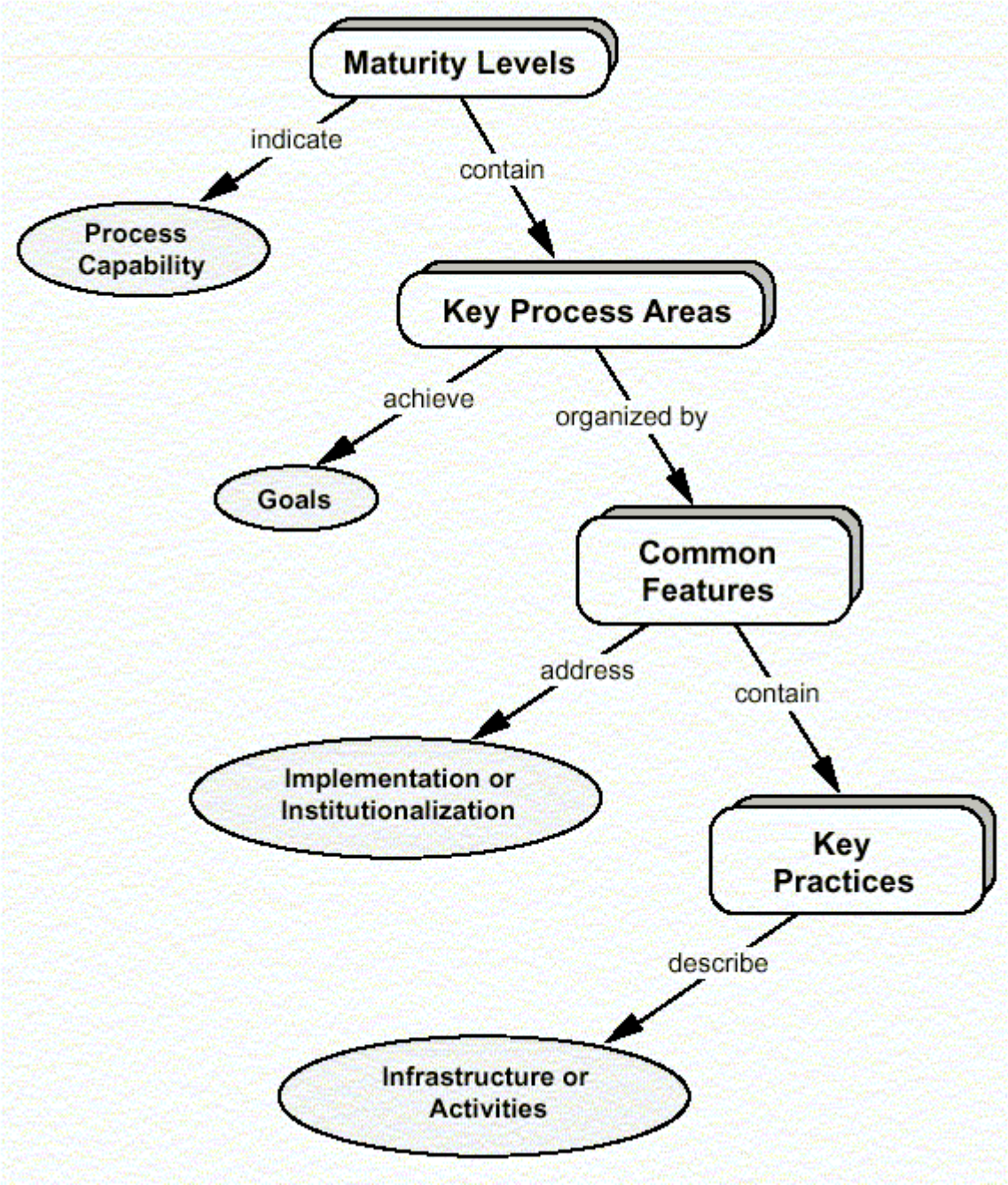
CMM Components

- Maturity levels
- Process capabilities
- Key process areas
- Goals and common features
- Key practices

©Joey Paquet 2000, 2002

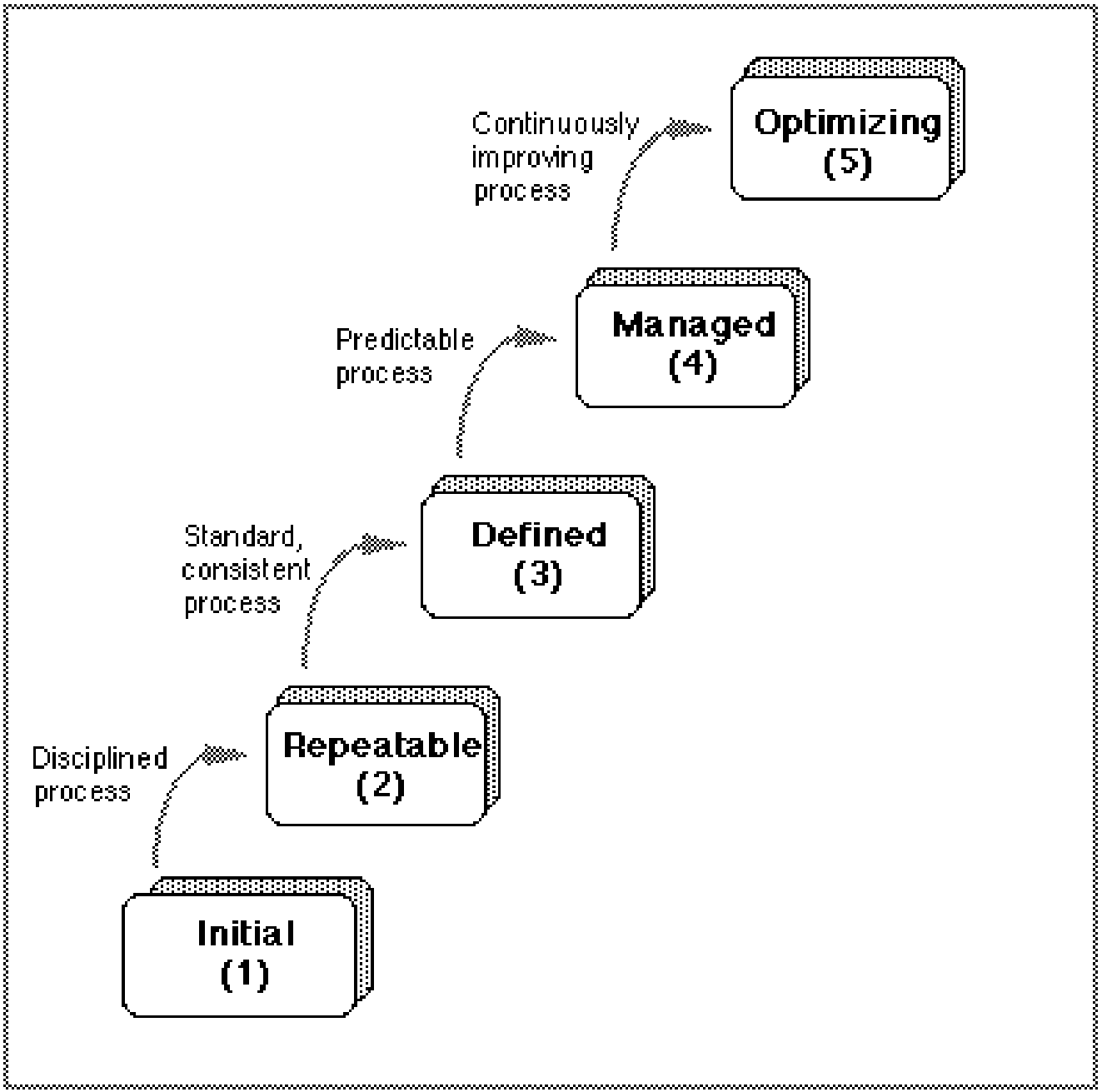
CMM

Components



©Joey Paquet 2000, 2002

CMM Maturity Levels



©Joey Paquet 2000, 2002

Level 1 - Initial

- Process is ad hoc, occasionally chaotic
- Few and informally defined processes
- No mechanism to ensure they are used consistently
- Ineffective planning
- Reaction-driven management
- Unpredictable
- *Success due to heroic efforts*
- ~80% of software organizations worldwide

©Joey Paquet 2000, 2002

Level 2 - Repeatable

- *Basic* management processes, quality assurance and configuration control procedures in place
- Can repeat earlier successes
- Realistic project commitments based on results of previous projects
- Still has frequent quality problems
- *Stable planning and tracking*
- ~15% of software organizations worldwide

©Joey Paquet 2000, 2002

Level 3 - Defined

- Documented, standardized, and integrated management and engineering processes
- Procedures are in place to insure they are followed
- Projects tailor organization's standard to develop own process
- Software Engineering Process Group (SEPG)
- *Stable foundation for software engineering and management*
- ~5% of software organizations worldwide

©Joey Paquet 2000, 2002

Level 4 - Managed

- Quality and performance is measured using metrics and can be predicted
- Quality and productivity *quantitative* goals are established
- Exceptional cases are identified and addressed
- Challenge of new domains can be managed
- *Process is measured and operates within limits*
- < 1% of software organizations worldwide

©Joey Paquet 2000, 2002

Level 5 - Optimizing

- Organization focuses on continuous process improvement
- Goal is to address and prevent problems by analyzing their cause in the process
- Process improvement is budgeted, planned, and part of the organization's process
- Identify and quickly transfer best practices
- Only a handful of organizations worldwide

©Joey Paquet 2000, 2002

Why Five Levels?

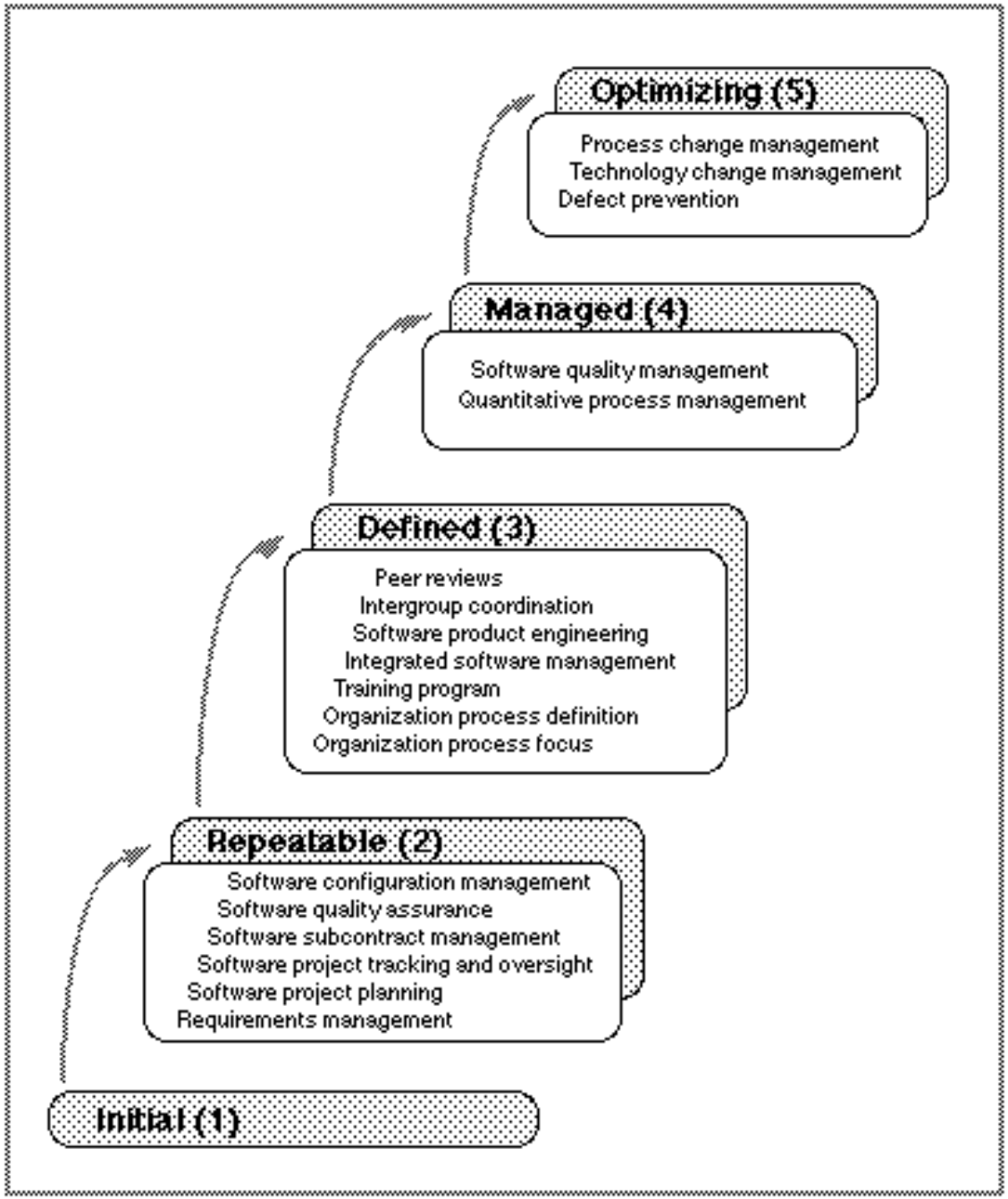
- Reasonably represents historical phases in actual software organizations
- Reasonable measure of improvement from previous level
- Suggests interim goals and measures
- Prioritize improvements

©Joey Paquet 2000, 2002

Key Process Areas (KPA)

- Issues to address to reach each maturity level
- Related activities for achieving goals

©Joey Paquet 2000, 2002



©Joey Paquet 2000, 2002

Key Process Areas—Repeatable(2)

- Requirements management
- Software project planning
- Software project tracking and oversight
- Software quality assurance
- Software configuration management
- Software subcontract management

©Joey Paquet 2000, 2002

Key Process Areas—Defined (3)

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

©Joey Paquet 2000, 2002

Key Process Areas—Managed (4)

- Quantitative process management
- Software quality management

©Joey Paquet 2000, 2002

Key Process Areas—Optimized(5)

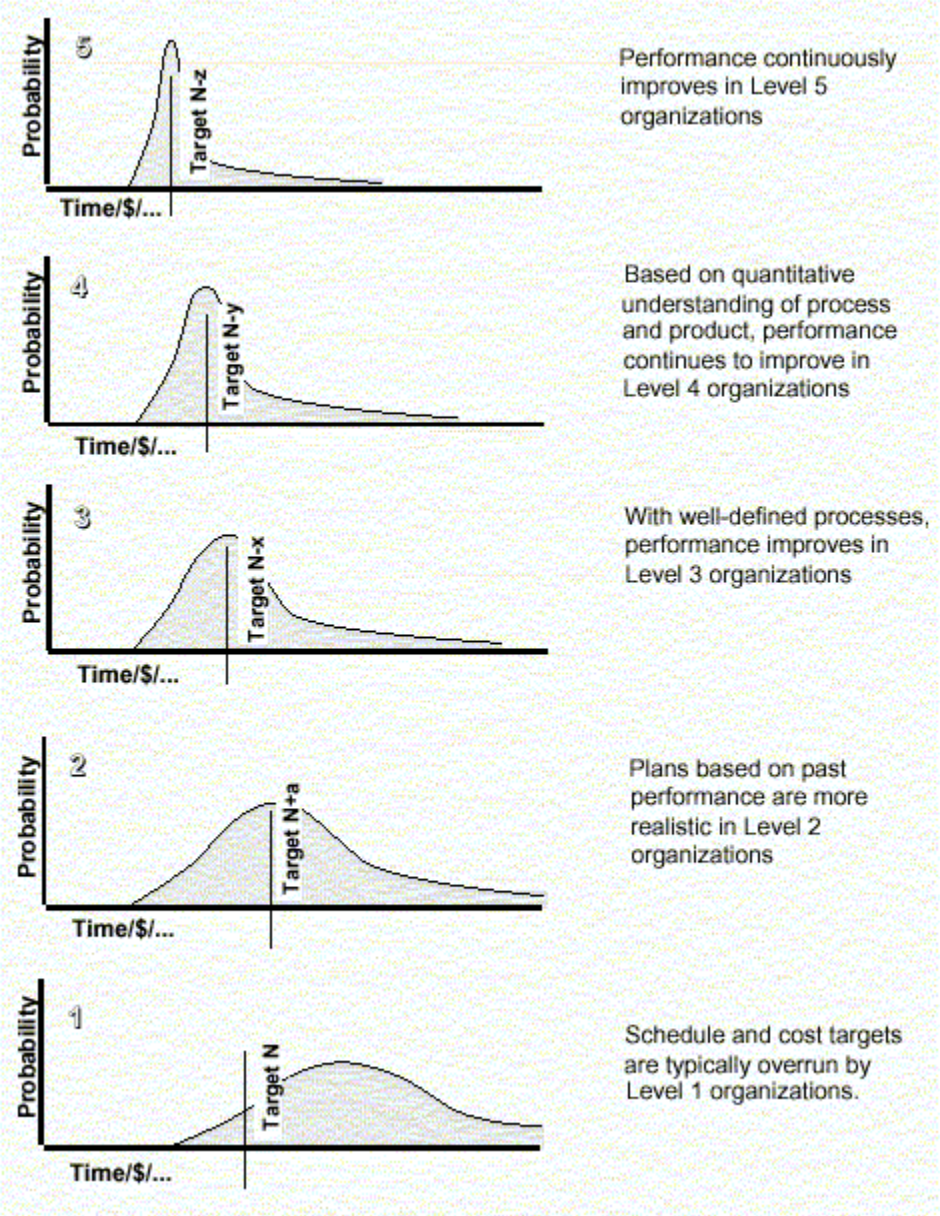
- Defect prevention
- Technology change management
- Process change management

©Joey Paquet 2000, 2002

CMM outcomes

- Fewer project overruns
- Better cost prediction (dollars, time)
- Predicted performance improvements

©Joey Paquet 2000, 2002

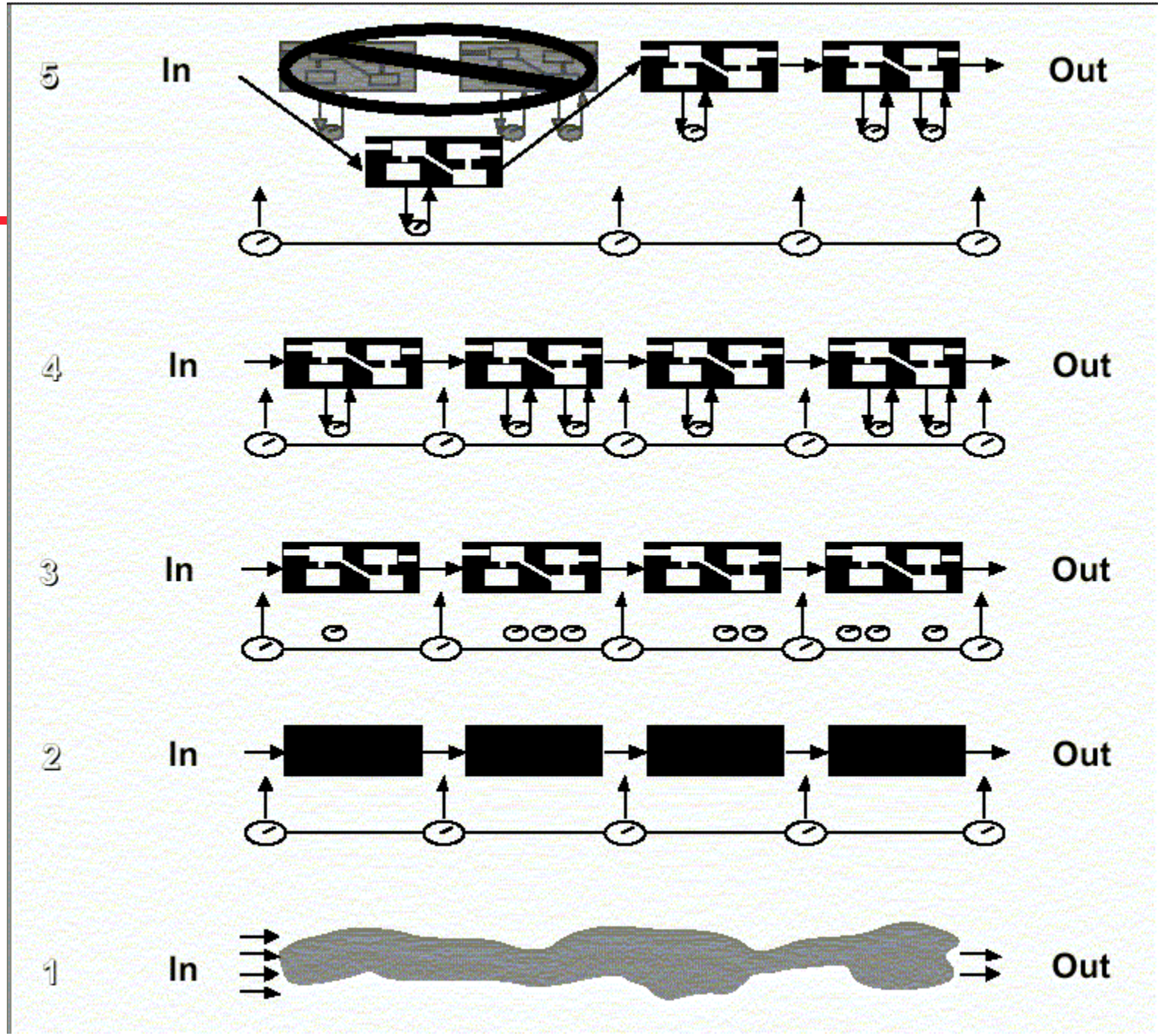


©Joey Paquet 2000, 2002

Visibility

1. No process internals regularly visible
2. Process allows a few visible check points
3. Process allows many regular visible check points
4. Process is quantitatively measured at many check points
5. Processes replaced by better processes

©Joey Paquet 2000, 2002



©Joey Paquet 2000, 2002

CMM in practice

"Although much is written about the topic in qualitative terms, little quantitative information is available. In many ways, the engineering process is an informational 'black hole' -- it draws in money and resources like a magnet but little data emerges." (Howard Rubin)

- Only 2 DoD contractors and one government organization have released data documenting their return on investment for software process improvement

©Joey Paquet 2000, 2002

CMM in practice

- Raytheon
  - 7.7:1 return on investment, savings of $4.48M
  - Elimination of $15.8M in rework costs
  - Twofold increase in productivity
- Hughes Aircraft
  - Initially Level 2, progressed to Level 3
  - ~$45K for assessment, ~$400K to improve
  - 5:1 ratio of ROI
  - Annual savings of approximately $2M
  - Decreased risk of missing cost and schedule estimates
  - Improved quality of work life (less overtime)
- Tinker Air Force Base
  - 6:1 ratio of ROI
  - Savings of $3.8M for $0.64M investment

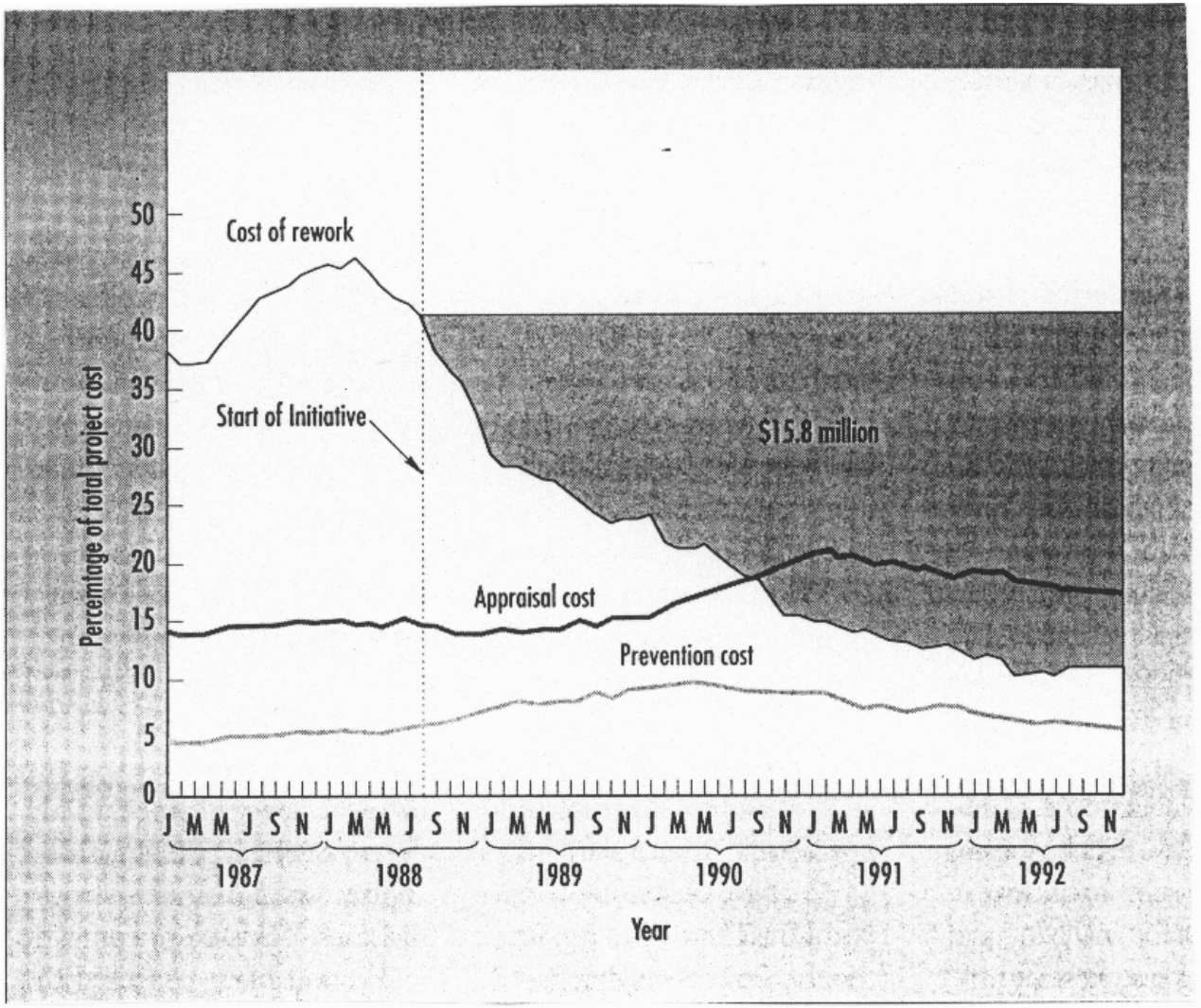
©Joey Paquet 2000, 2002

Validity of Published Data

- What's the baseline?
- Is the observed process improvement attributable to the CMM?
- Was the data obtained by comparing "Apples and Oranges"?

©Joey Paquet 2000, 2002

Savings at Raytheon



©Joey Paquet 2000, 2002

Savings at Raytheon

- Cost increases and decreases
  - Fixing defects during design: 2.5 times increase
  - Fixing defects during coding: 1.75 times increase
  - Integration cost: decrease by 80%
  - Retesting cost: decrease by 50%

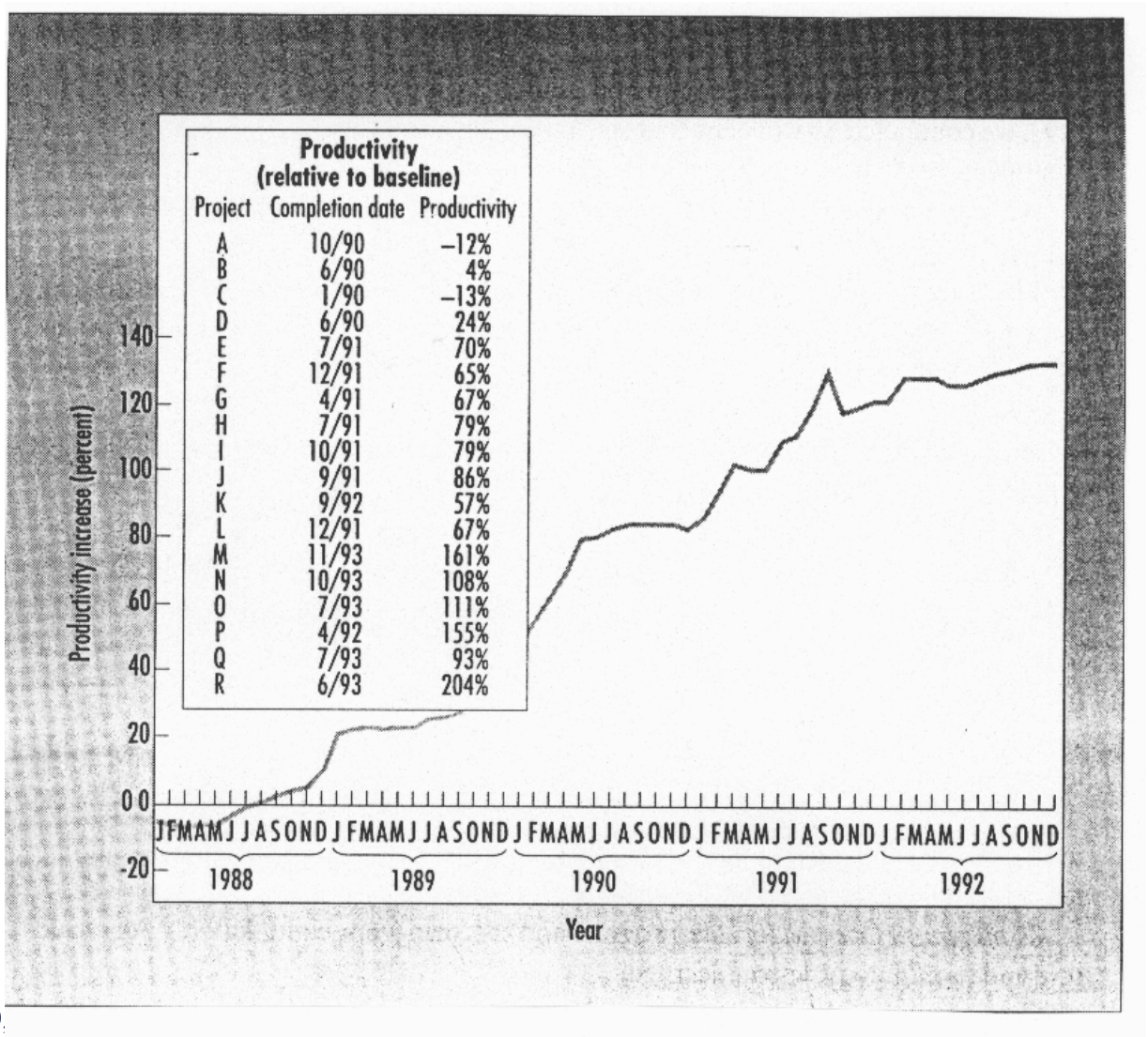
©Joey Paquet 2000, 2002

Raytheon Productivity Data

- Measure: equivalent delivered source instructions per man-month
- Modified and reused LOC are weighted according to the relative effort of modification or reuse compared to new code
- Not scientifically accurate because of variations among the projects

©Joey Paquet 2000, 2002

Raytheon Productivity Data



©Joey Paquet 2000

Data from a wider range of US Industry

- Data from 35 companies and government agencies
- various levels of maturity
  - mostly Level 1-3, all organizations at Level 4 and 5 participated
- various geographical locations

©Joey Paquet 2000, 2002

Cost of Process Improvement

Category	Measurement	Measured Cost Increases
Dollars	SW and HW costs	5% increase
Efforts	data collection cost	increased to 5-10% of effort
	cost of fixing design defects	increased from 0.75% to 2% of project cost
	cost of fixing code defects	increased from 2.5% to 4%
	cost of first-time testing	increased to 9-10% of effort
	inspection overhead	increased to 2% of development time

©Joey Paquet 2000, 2002

Drawbacks of CMM

- CMM is not a “Silver Bullet”
- What do you think some drawbacks are?

©Joey Paquet 2000, 2002

Drawbacks of CMM

- Lack of innovation?
  - CMM says what you need, not how to do it
  - Usually because processes done wrong
  - Processes are supposed to routinize mundane aspects of work so you can focus on interesting aspects
  - Do it right, and do it right once
  - Even Microsoft has well defined processes, estimates itself to be at Level 3

©Joey Paquet 2000, 2002

Drawbacks of CMM (cont.)

- Two common complaints about CMM
  - Does not focus on good design
  - Does not focus on good people
  - However, these are not the goals of CMM!
- Other models have been developed
  - P-CMM (People Capability Maturity Model)
  - Team Capability Maturity Model
  - PSP - Personal Software Process

©Joey Paquet 2000, 2002

Drawbacks of CMM (cont.)

- Describes what an organization should have, does not say how to get there
- Clearly defined process != good process
- Favors narrow maintenance processes over innovative ones
- Process Assessment flaws
  - Statistical problems, sparse coverage, process risk

©Joey Paquet 2000, 2002

Drawbacks of CMM (cont.)

- Government may require at least Level 3 compliance for contractors
  - Private industry likely to follow suit
  - May lead to political problems
- May lead to ossification of software processes

©Joey Paquet 2000, 2002

Drawbacks of CMM (cont.)

- Assembly-line process and control concepts are applicable to software
  - In manufacturing, majority of costs and risks are in replication
  - In software, majority of costs and risks are in design
  - Almost exact opposites!
- Even though in use, *still unproven*

©Joey Paquet 2000, 2002

ISO 9000

- International Standard Organization (ISO) – ISO 9000 (1987, 1991)
- A series of 5 related standards that are applicable to a wide variety of industrial activities.
- Used for a wide range of industrial applications. (therefore not certainly a software standard)
- ISO 9001 Standard for quality system (closest related to software engineering).
- Like the CMM the ISO 9000 emphasizes measurement. Both models stress on documenting the process in word and pictures.
- It has been reported that at least two level 1 organization have been certified as compliant with ISO 9000.

©Joey Paquet 2000, 2002

SPICE

- Spice is intended to extend and improve the CMM and ISO 9000 models.
- Differences include that Spice provides a framework for assessment methods, but does not lay down a specific method.
- Spice provides a separate assessment of each component of the overall process (analysis, specification, configuration management, etc…)

©Joey Paquet 2000, 2002

Conclusions

- Good processes are essential
- We are still learning about good processes for software development
- CMM was developed to assess and to give organizations a framework to improve
- Despite some flaws, CMM is a significant contribution to the software industry
- CMMv2 in progress at http://www.sei.cmu.edu/

©Joey Paquet 2000, 2002