

Managing e-Business/ Internet-Time Projects









Ed Yourdon

ed@yourdon.com

<http://www.yourdon.com>








version 2.05, May 2001

Agenda

-  Introduction and quick summary
-  Project Politics
-  BPR for e-Business projects
-  Project Negotiations
-  Peopleware Issues
-  Software Processes
-  Monitoring and Controlling Progress
-  Languages, Tools, and Technology

INTRODUCTION :










What's so different about e-business projects?

-  ***Users and managers are becoming ever more demanding.*** Many e-business projects are defined as “mission-critical”
-  ***Many e-business projects require BPR to succeed*** — just like the early days of client-server projects, during which we learned that 80% of BPR projects were failures.
-  ***Peopleware issues are often exacerbated*** — shortage of skills, decreasing company loyalty, inclusion of people with no formal IT background (“testing? what’s that?”)
-  ***Pace of business demands ever-faster implementation.*** Death-march project schedules tempt project teams to abandon all “process”
-  ***“External” e-business projects are often exposed to much greater risks than before*** — e.g., performance problems, reliability, security, privacy, etc.
-  ***New technologies are emerging ever faster ...*** e.g., wireless technologies, Microsoft’s .NET, new development environments, etc.
-  For more details, see my Aug 2000 *Computerworld* article, “[What’s So Different About Managing E-projects?](#)”

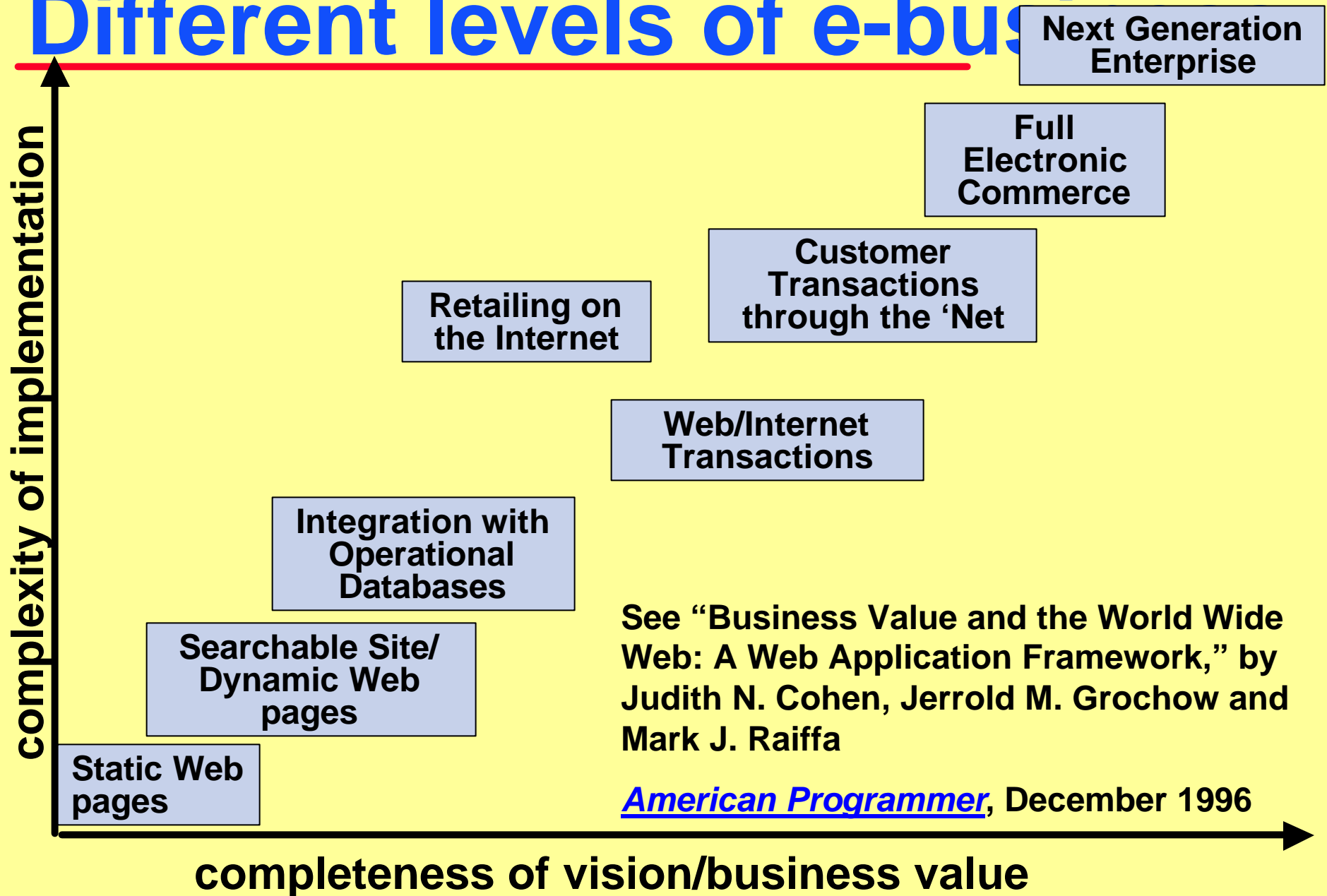
INTRODUCTION:

What's so different about e-business projects?

Users and managers are becoming ever more demanding

-  Recognize that some of this is caused by the “hype” associated with e-business and the Internet
-  Recognize that the hype is beginning to fade now that (a) e-business has been around for a while, and (b) many dot-com companies have collapsed, and (c) many large companies have cut back on their e-business plans. See [“CFOs on e-business: not so fast,”](#) by Thomas Hoffman, *Computerworld*, Mar 1, 2001)
-  This puts more emphasis than ever on the “politics” of projects
 -  Who has the political power to declare project success or failure?
 -  Who are the other key stakeholders in the project?
 -  What are the key criteria for success?
-  The “visible” aspects of success may involve an aggressive deadline, but “silent” issue is likely to involve reengineering business processes
 -  Hence, more emphasis than ever on BPR
 -  And more emphasis than ever on project negotiations

Different levels of e-business



E-commerce business issues







- ✍ **Creates global, 24-by-7 business environment**
 - ✍ Provides both advantages and disadvantages
 - ✍ But no longer optional for all but the smallest business
- ✍ **Forces companies to reconsider fundamental questions**
 - ✍ What business are we in?
 - ✍ What is our core competence?
 - ✍ What basic assumptions, constraints are relevant?
- ✍ **Provides customers with far more power than ever before — and thus changes traditional relationship with suppliers**
 - ✍ Most companies don't have a clue about what this really means
 - ✍ Example: if you run company X, look up “I hate X” or “X sucks” in your favorite search engine
 - ✍ For more (provocative) details, see [The Cluetrain Manifesto](#)



INTRODUCTION:

What's so different about e-business projects?










Many e-business projects require BPR to succeed

-  We had the same experience a decade ago, when client/server technology was introduced
-  BPR guru Michael Hammer concluded that 80% of BPR projects were failures.
-  Many e-business projects were launched with no “business model” for identifying benefits, revenues, customers, or rationale for success
-  Many e-business projects are “re-paving old cowpaths,” using Internet/web to carry out the same old business processes with new technology, but no fundamental improvements.
-  Thus, e-business project managers must realize that part of their role is be the BPR “champion” or “facilitator”
-  But there is a big risk: most project managers lack the authority or political power to impose BPR upon user departments.

INTRODUCTION:

What's so different about e-business projects?






Peopeware issues are often exacerbated

-  **Typical problems:**
 -  shortage of skills,
 -  decreasing company loyalty,
 -  inclusion of people with no formal IT background (“testing? what’s that?”)
-  **Skill shortage is less serious with collapse of dot-com industry, but it’s still hard to find people with advanced skills**
-  **Dot-com collapse has made many employees more cautious, less willing to take low salaries + stock options**
-  **Economic slowdown has exacerbated many peopeware problems (e.g., budget cuts have eliminated funds for training)**
-  **High-pressure “death-march” e-business projects create more peopeware problems.**
-  **No “instant solution” to these problems, but it’s more important than ever to be aware of peopeware issues and strategies.**

INTRODUCTION:

What's so different about e-business projects?





Pace of business demands ever-faster implementation

-  **Death-march project schedules tempt project teams to abandon all “process”**
-  **Similar phenomenon occurred in early 90s, when client/server projects were first introduced**
-  **E-business project teams often composed of young software engineers, often without any formal software engineering training**
-  **But some e-business projects are now large and complex enough that “formal” methodologies are accepted as necessary**
-  **For small, fast e-business projects, current strategy is to use “light” or “agile” methods as a compromise between “bureaucratic paralysis” and “anarchy”**

INTRODUCTION:

What's so different about e-business projects?







“External” e-business projects are often exposed to much greater risks than before

-  Typical problems: performance problems, reliability, security, privacy, etc.
-  Examples: Hollywood Oscar [web site](#), [Olympics](#), launch of [iGrandparents.com](#) web site (covered on national evening news)
-  Puts more emphasis than ever on careful planning and testing
-  Also puts more emphasis than ever on risk management as a significant part of project planning and project management

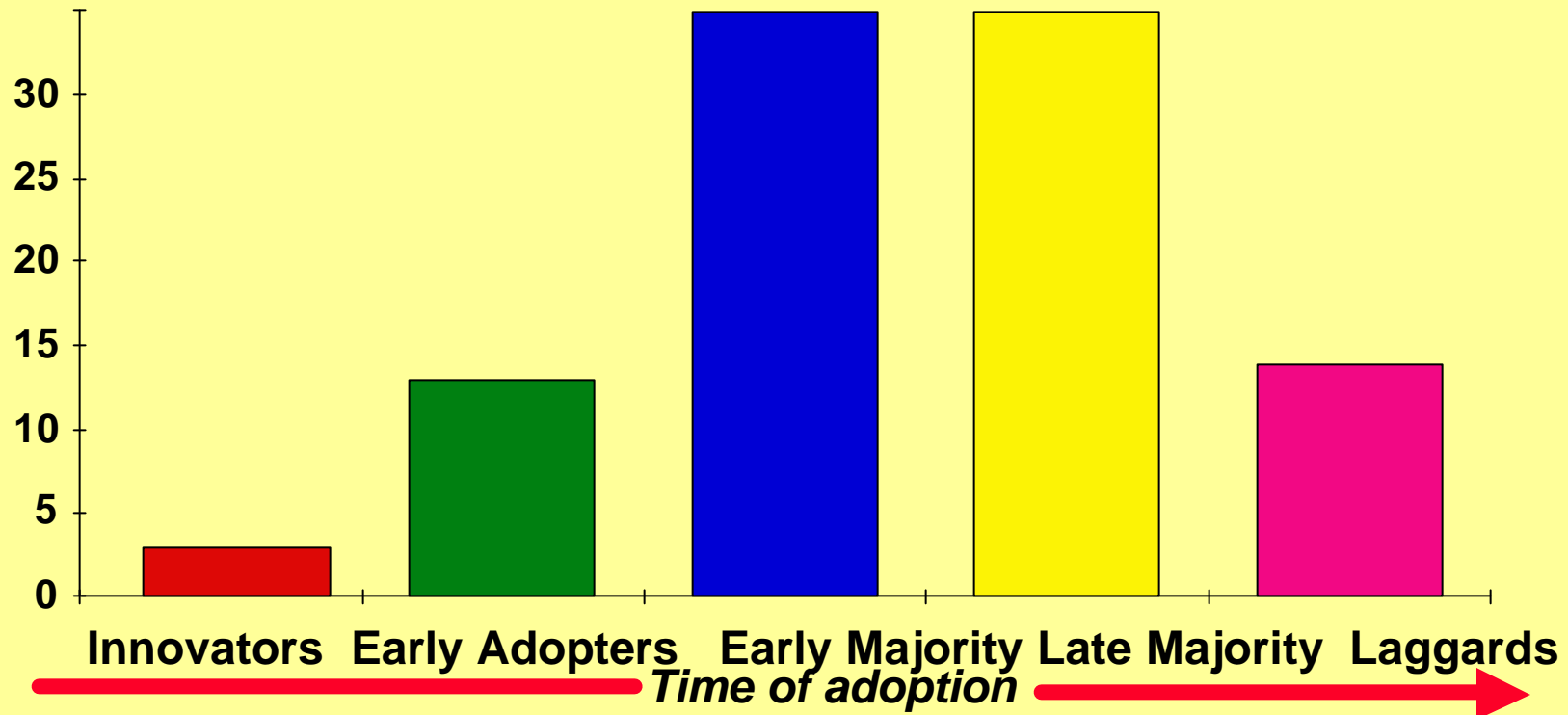
INTRODUCTION:

What's so different about e-business projects?

New technologies are emerging ever faster

-  **Examples: wireless technologies, Microsoft's .NET, new development environments, new versions of Java, handheld devices, etc.**
-  **But this has been true for each of the “waves” of technology that have swept through the IT industry**
-  **It can obviously be a problem if your existing IT staff doesn't have these skills...**
-  **... or if the technologies are so new that they don't work reliably, or are not supported properly by the vendors, etc.**
-  **But my experience has been that new technologies are not the fundamental cause of project failures; sometimes they are the “straw that breaks the camel's back.”**
-  **Key management decision: do you want to be an “innovator,” an “early adopter,” a “mainstream” technology user, or a “laggard”?**

Technology adoption segments

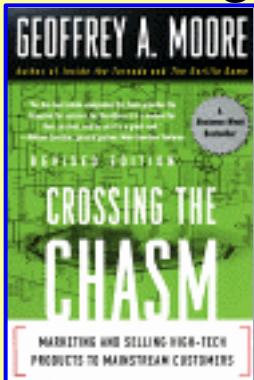


—— Strategic objective ——

— Technological imperative —




—— Value added ——

— Cost Displacement/Avoidance —











see Geoffrey Moore, [Crossing the Chasm](#)

2. PROJECT POLITICS





-  **Identifying owners, customers, shareholders, and stakeholders in an e-business software project**
-  **Determining the basic nature of the project**
-  **Levels of commitment**


2.1 Identifying the players

-  **Key point: your chances of success in an e-business project are often zero if you don't know who the key players are**
-  **Also crucial that everyone on the project understands this — not just the project manager**
-  **Typical players:**
 -  **Owner: who pays for, or “accepts” the system?**
 -  **Customer: who will actually use the system?**
 -  **Shareholder: one of many “co-owners”**
 -  **Stakeholder: someone whose interests are affected by the success/failure of the project, and who can affect the outcome of the project — and who may thus become an ally or obstacle to project success. See “Project Clarity Through Stakeholder Analysis,” by Larry Smith, [Crosstalk: The Journal of Defense Software Engineering](#), Dec 2000**
-  **Note: because e-business projects can change the way the business is run, the “traditional” identification of owners, customers, stakeholders may be inaccurate.**

2.2 Determining the basic nature of the e-business project

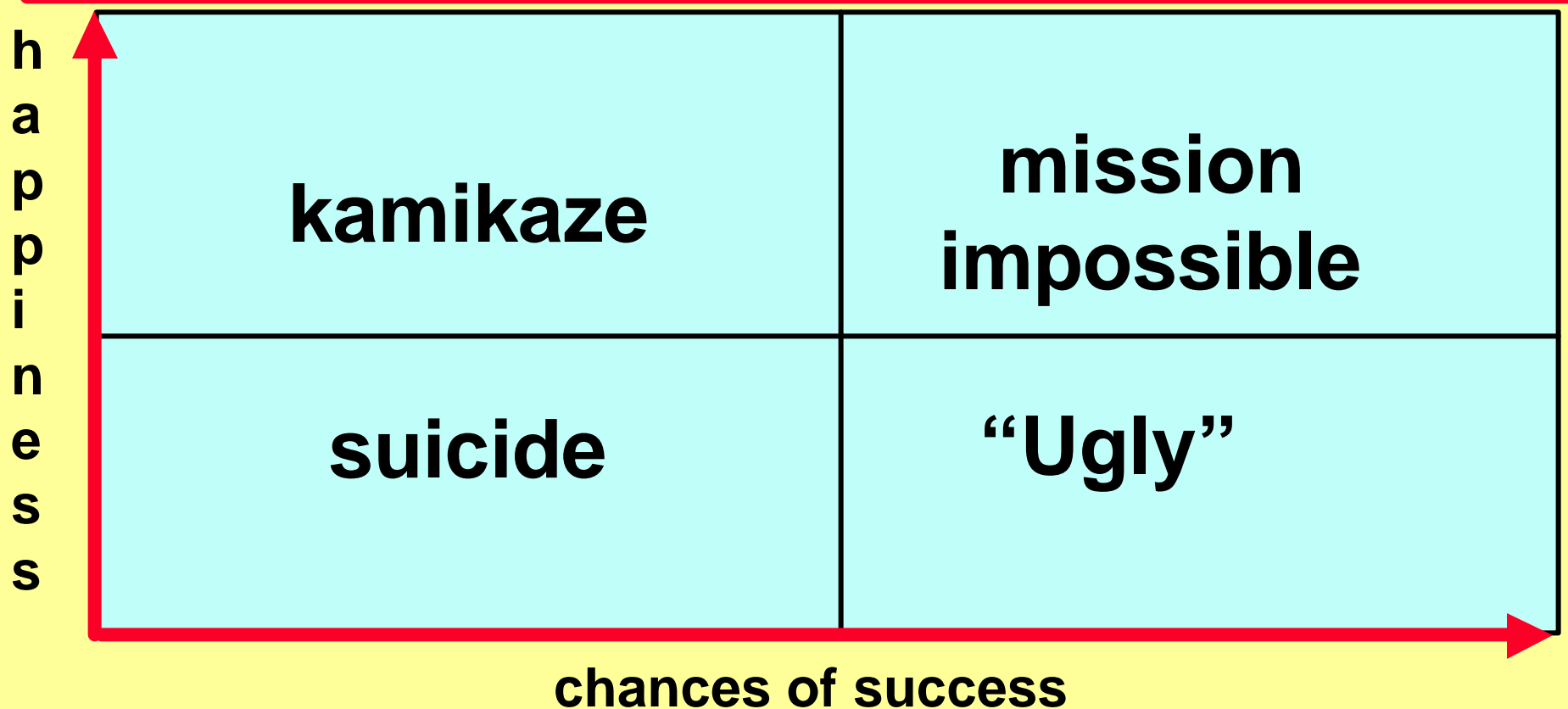
Typical examples:

-  mission impossible: if we succeed, we live happily ever after
-  kamikaze: the project may succeed, but it will kill all of us
-  ugly: the project manager is prepared to sacrifice any and all of the team members in order to succeed.
-  suicide: the project has no chance of success, and we're the scapegoats

 Remember that public assurances from executive sponsor may not reflect the “reality” of the situation





 Also remember that the situation can change dynamically, based on politics, external business conditions, gov't regulations, etc.

2.2 Determining the Basic Nature of the Project









Key point: get the project team members to indicate where they think *they* fit into this grid.

2.3 Levels of Commitment

-  **The parable of the chicken and the pig**
-  **Remember that different members of the “key players” have different levels of commitment...**
-  **...some of which may be publicly stated, and some of which may not**
-  **...and all of this may change on a daily basis**










3. Business Process Reengineering

-  Introduction
-  The Role of IT in a BPR Project
-  Reengineering the IT Organization
-  Critical Success Factors in BPR
-  A BPR Management Plan
-  An Action Plan for Getting Started

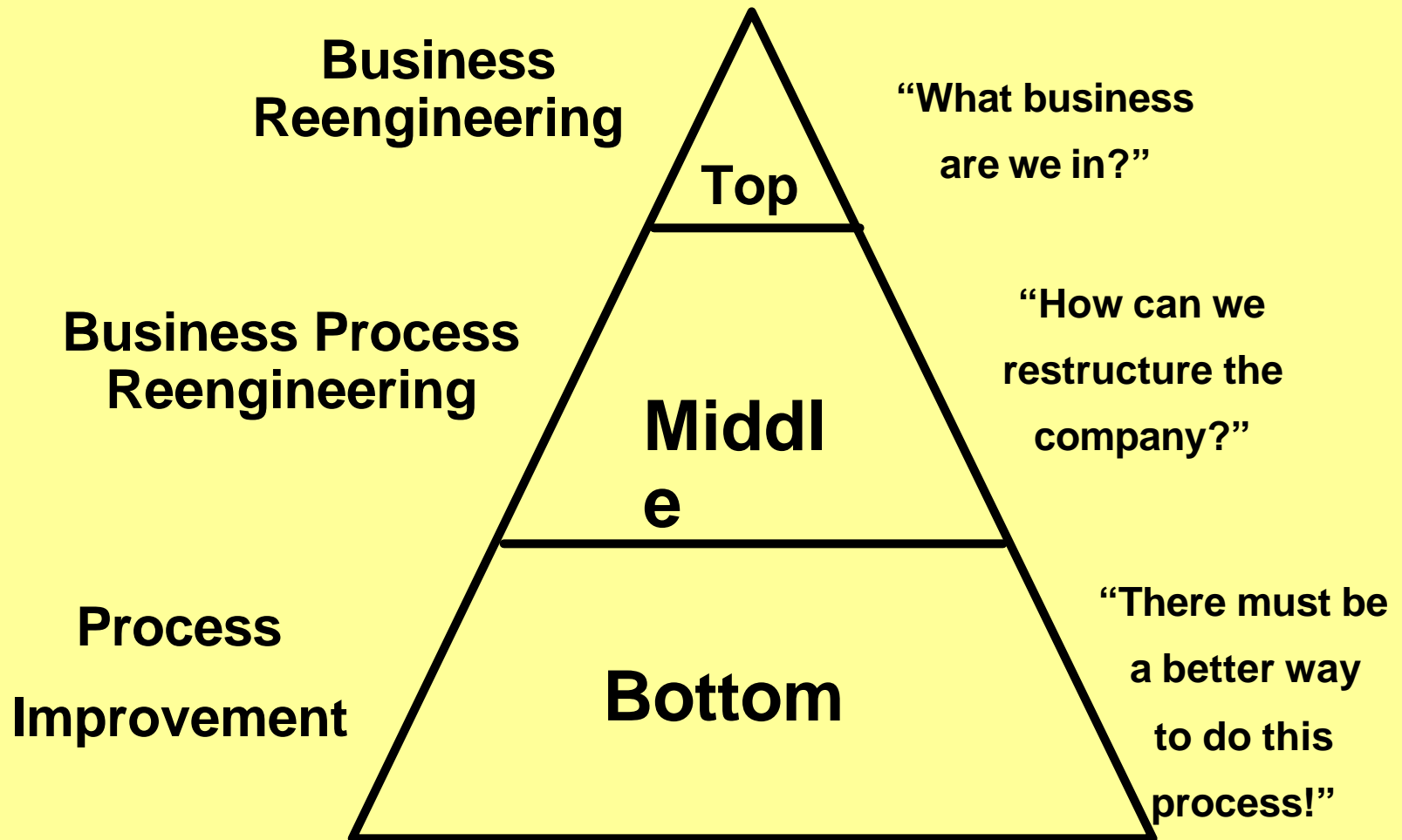
3.1 Introduction — what is BPR?

Business reengineering is the fundamental rethinking and radical redesign of an entire “business system”—the business processes, jobs, organizational structures, management systems, and values and beliefs—to achieve dramatic improvements in critical measures of performance

Why does BPR happen?

-  **Primarily because of extreme pressure for improvement**
-  **“Extreme pressure” is usually initiated externally**
-  **Typical factors**
 -  **intense competition in global markets**
 -  **radical opportunities caused by new technologies (e-business)**
 -  **social/political change (e.g., privatizing of public enterprises)**
-  **Typical private sector example: discovering your competitor generates same revenue with much less people**
-  **Another example: discovering your competitor has a process with “cycle time” much faster than yours (e.g., processing time for a mortgage application in a bank, or response time for customer service inquiry)**
-  **Typical public sector example: taxpayer revolt that leads to 50% reduction in budget for a government agency — or politician’s promise of “e-government”**







Where does BR/BPR originate in the organization?



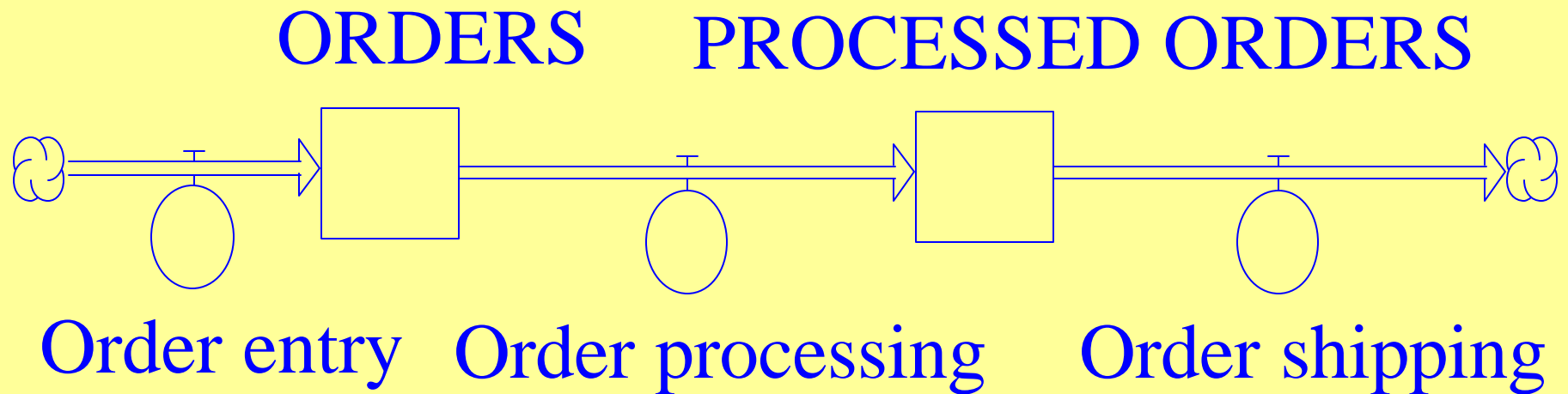
What is a “process”?

- ✍ Not a department, or a person, or a computer, or a “functional area,” etc.
- ✍ “A process is simply a structured, measured set of activities designed to produce a specified output for a particular customer or market.” (Tom Davenport, [*Process Innovation: Reengineering Work through Information Technology*](#))
- ✍ “A process is thus a specific ordering of work activities across time and place, with a beginning, an end, and clearly identified inputs and outputs: a structure for action.” (Tom Davenport, [*Process Innovation: Reengineering Work through Information Technology*](#))
- ✍ “A collection of activities that takes one or more kinds of input and creates an output that is of value to the customer.” (Michael Hammer and James Champy, [*Reengineering the Corporation*](#))
- ✍ For IT people: a DFD bubble, or an object

More on processes

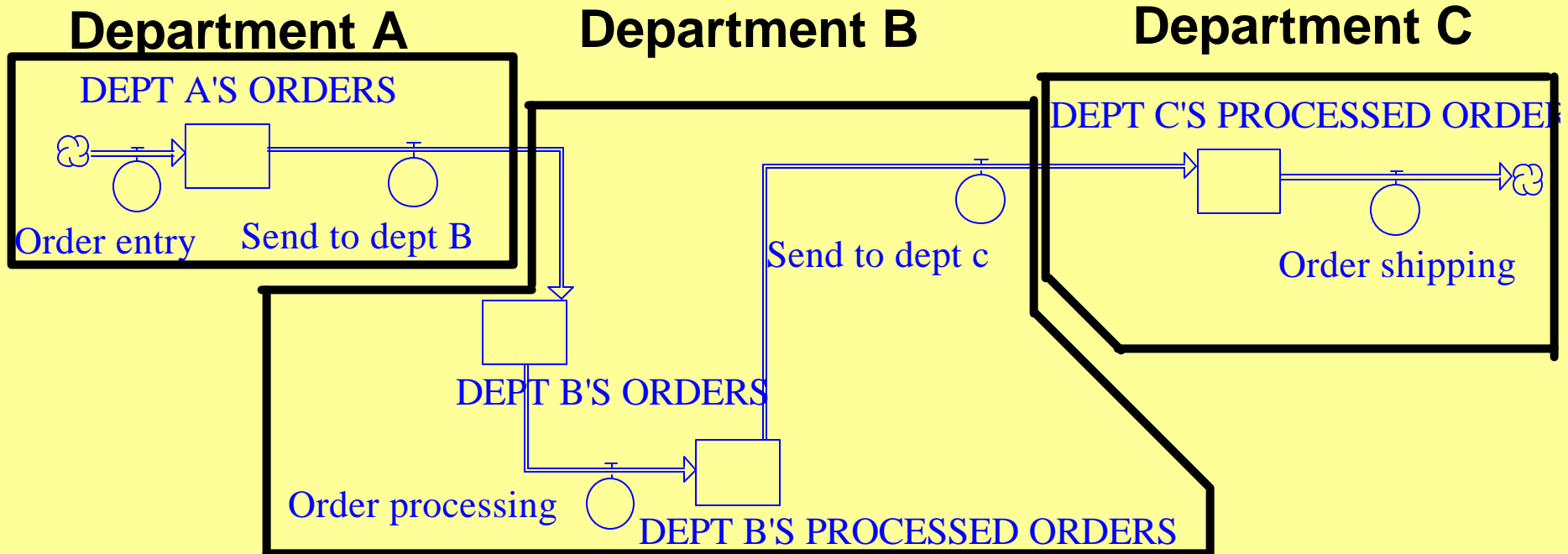
-  Processes can be big or small...
-  Processes can contain embedded “sub”-processes
-  So an organization may choose to identify hundreds (thousands) of “little” processes...
-  ...or a much smaller number of “core” or “basic” processes that characterize the entire enterprise.
-  Improving tiny processes is low-risk, but will usually have only incremental improvements — thus, it’s not usually the focus of an e-business project
-  Focusing on “core” processes often looks for major improvements in the “hand-off” between processes

Examples of “hand-off” problems



- ✍ From the customer’s perspective, the process may be as simple as this — and in many cases, the customer would prefer not to spend any time talking or interacting with corporate employees to accomplish the process.
- ✍ But organizational “hand-offs” between departments can complicate the situation.

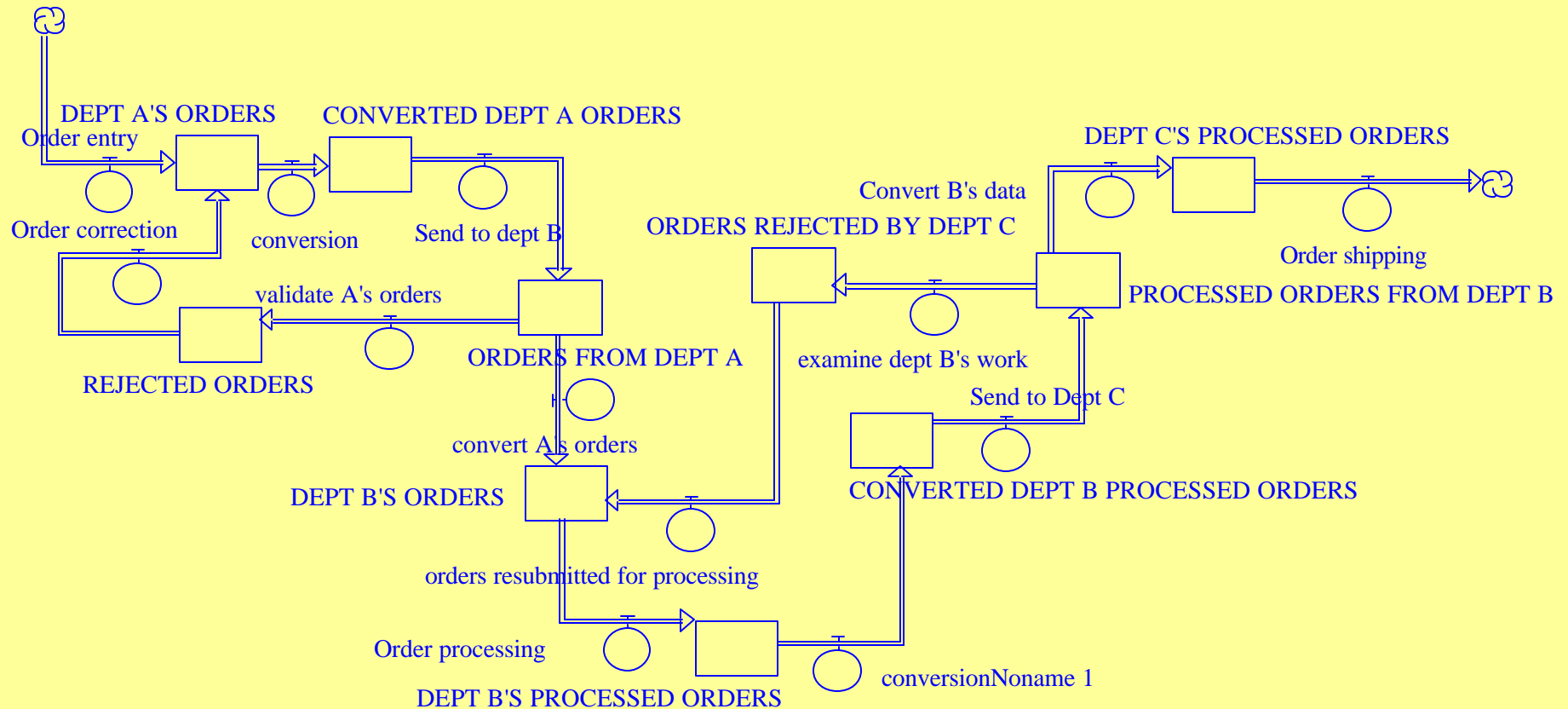
Examples of “hand-off” problems



✍ But since Department A, B, and C each have their own computer systems, the data moving between the departments has to be “converted”











✍ And there may be data errors, which have to be sent “upstream” for correction...

Examples of “hand-off” problems



- ✍ Situation can be made even worse by “inquiry” processes to track things that are lost between departments
- ✍ You can make each “bubble” more efficient, but the real problem is the “hand-off” between the original “core” processes

Common “core” processes

-  **Rockart and Short identify three:**
 -  developing new products
 -  delivering products to customers
 -  managing customer relationships
-  **Harvard researchers argue there are only 2:**
 -  managing the product line
 -  managing the order cycle
-  **IBM identified approx 140 processes across the organization in the ‘80s, but simplified it to 18 broader processes in the ‘90s**
-  **Most large organizations end up with approx 10-20 “core” processes that they focus on.**
-  **But it’s hard to know what the right ones are until you’ve modeled all the detailed processes.**

Common “core” processes

IBM (18)

Market info capture
Market selection
Requirements
Development of hardware
Development of software
Development of services
Production
Customer fulfillment
Customer relationship
Service
Customer feedback
Marketing
Solution integration
Financial analysis
Plan integration
Accounting
Human resources
IT infrastructure

Xerox (14)

Customer engagement
Inventory mgmt & logistics
Product design and devlpmnt
Product maintenance
Technology management
Production and operation
Market management
Supplier management
Information management
Business management
Human resource mgmt
Leased & capital asset mgmt
Legal
Financial management

British Telecom (15)

Direct business
Plan business
Develop processes
Manage process operations
Provide personnel support
Market products & services
Provide customer service
Manage products & services
Provide consultancy service
Plan the network
Operate the network
Provide support services
Manage information svcs
Manage finance
Provide technical R&D

from Tom Davenport's, [Process Innovation: Reengineering Work through Information Technology](#)

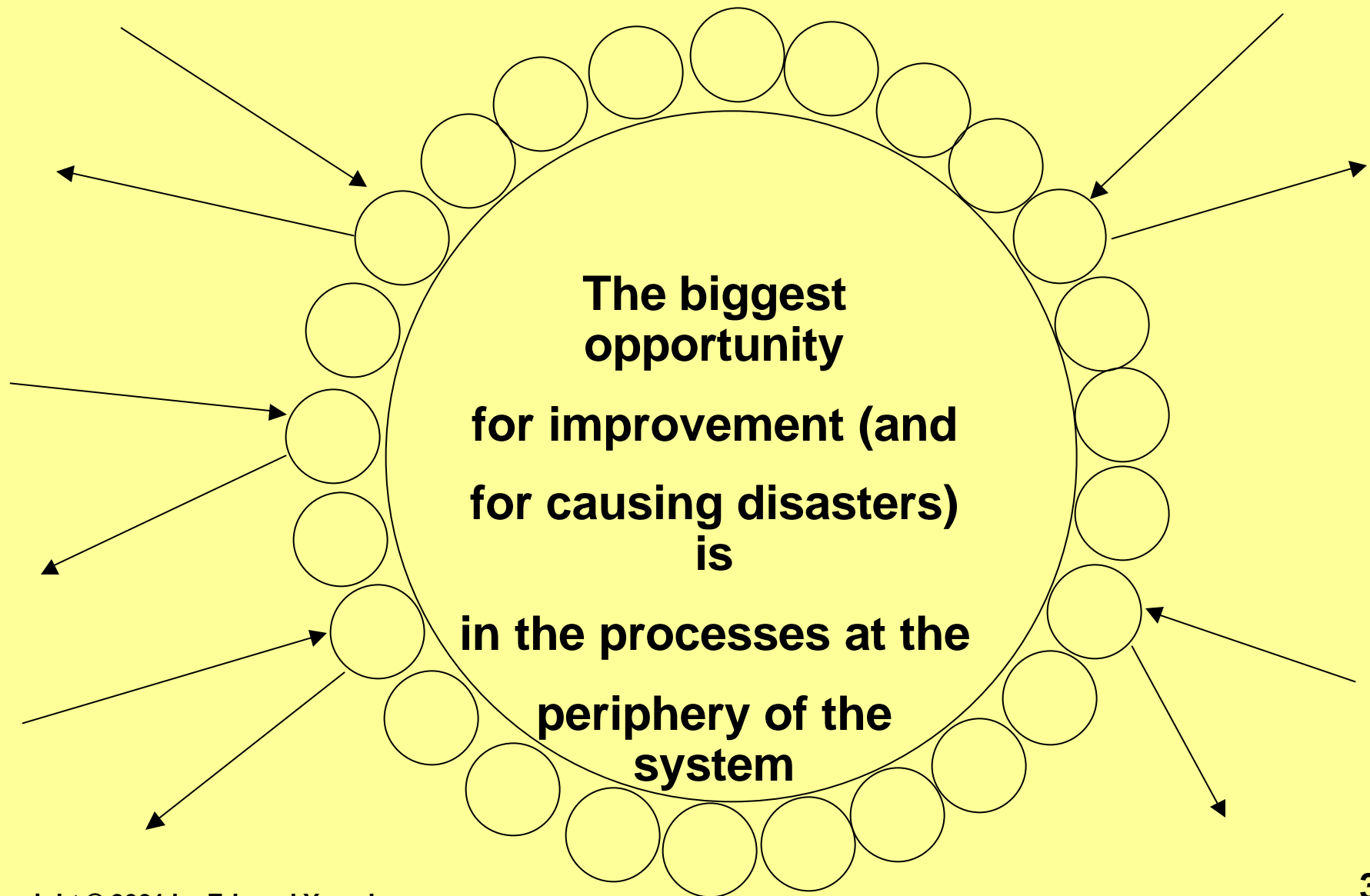
Note: What value does “legal” have to the customer in the Xerox list?

Fundamental BPR concepts

- 1. The objective of a business or enterprise is to add value, and core business processes are the mechanism by which value is added.**
- 2. In every business, there is *at least* one core business process that represents the main-line flow of business activity**
- 3. Business process reengineering attempts to clear away the underbrush of organizational history, clear/rebuild the main arteries of the system , and focus on the ultimate customer.**

**Ken Orr, “How Real is Business Process Reengineering Really?”,
American Programmer, November 1993**

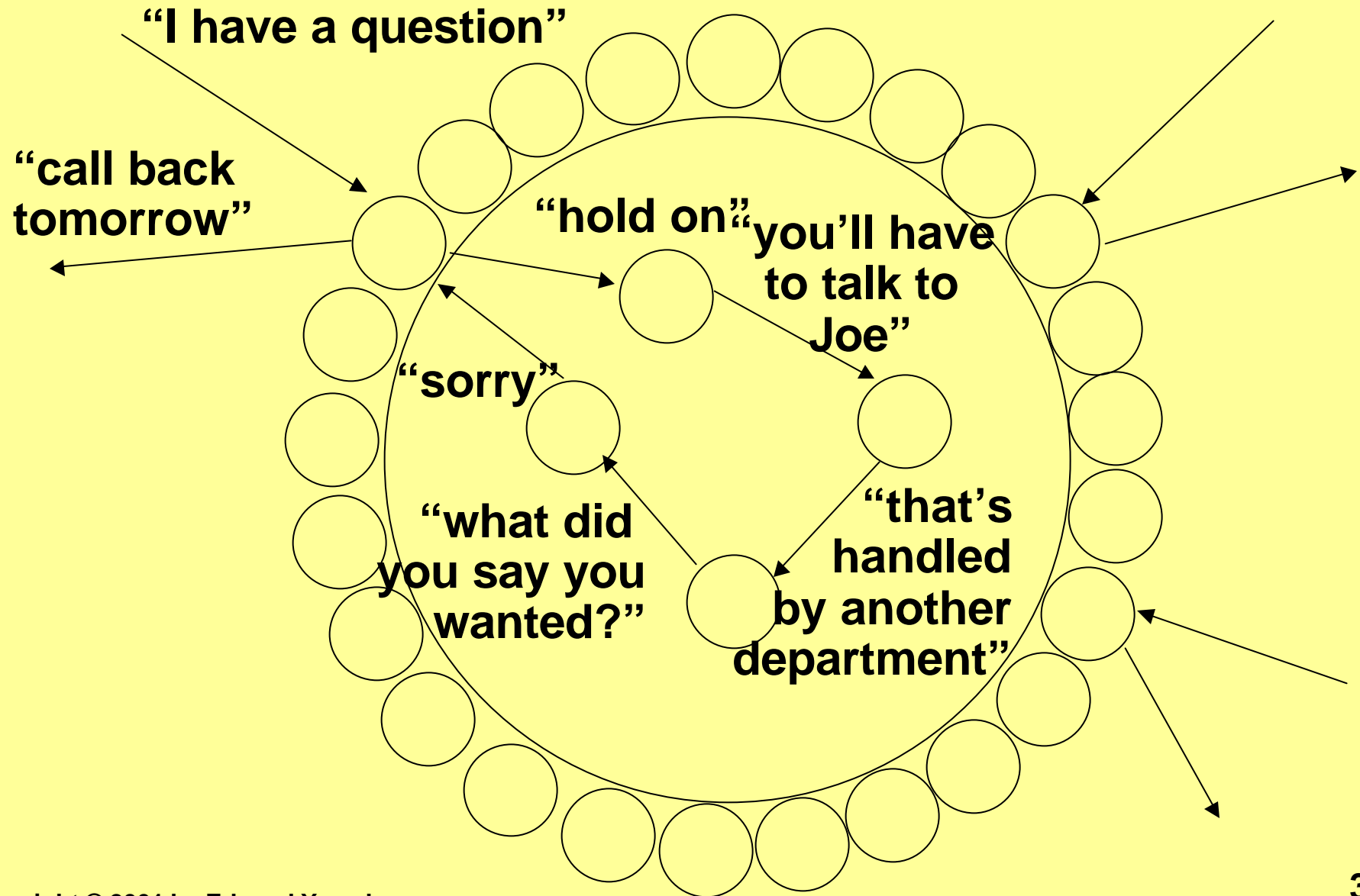
The focus of many BPR projects







Questions to ask

- 1. How long does process take? How many errors?**
- 2. How many “handoffs” does customer have to endure with other processes in the organization?**
- 3. How many more times does the customer have to call back to get his transaction carried out?**
- 4. What percentage of “top ten questions” can be answered by customer service person first time?**
- 5. What percentage of “top ten objections” can be answered by customer service person first time?**

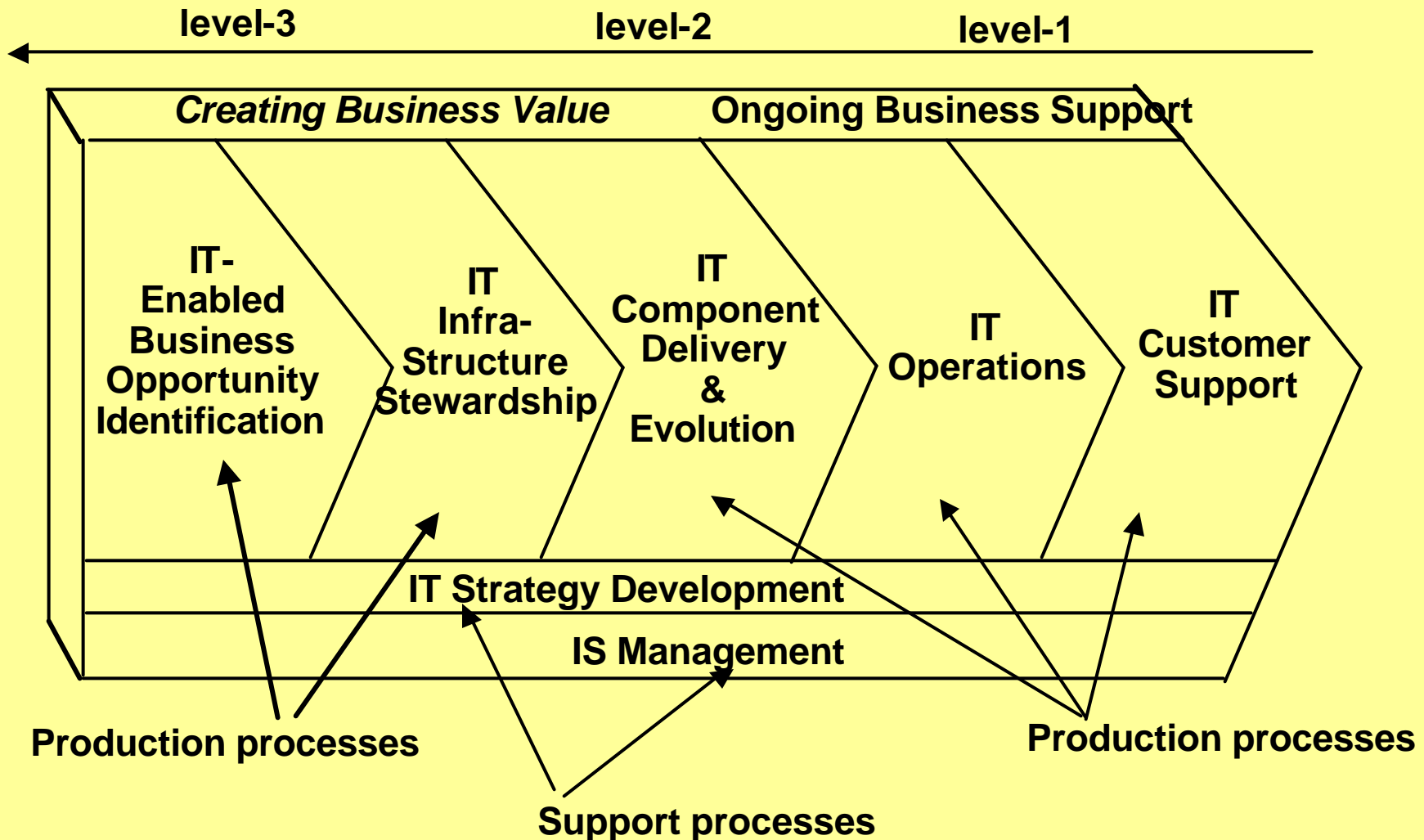
A common situation



3.2. The Role of IT in e-Business BPR Projects

-  **Can IT lead a BPR project?**
-  **Politics and realities**
-  **Suggestions for a practical approach**
-  **Recommended role of IT**

The E&Y IS Leadership Model






See "Attaining Top-Level IS Performance," by Mary Silva Doctor and Richard W. Swanborg, Jr., Ernst & Young, Jan 1994










The role for IT in BPR

- ✍ “Level-3” IT organizations have earned the credibility to initiate BPR projects, based on IT opportunities they can see. Level-3 IT organizations have a process model for their own organization!
- ✍ “Level-2” IT organizations are typically regarded as a “partner” in the BPR, along with user groups
- ✍ “Level-1” IT shops need to concentrate on getting their own house in order; users don’t trust them, and perceive them as part of the problem
- ✍ If you don’t know where you are on the scale(!), ask senior management—or get an objective, credible outside assessment.
- ✍ For articles on reorganizing IT for e-business, see the January 2001 issue of [Cutter IT Journal](#). Also, see “IT’s Role in Transitioning to E-business,” by Geoff Dober, [Cutter Consortium Business-IT Strategies Advisory Service](#), Vol. 4, No. 1, and “Ensuring IT is E-Business Ready,” by Ian Hayes, [Cutter Consortium Business-IT Strategies Advisory Service](#) Executive Report, Vol. 3, No. 4






The role for “level-1” IT shops

-  **Main role: working with user departments to modeling the business process, whether it is currently automated or manual (or both)**
-  **Process modeling (e.g. data flow diagrams) is more popular than data modeling**
-  **CASE tools can play an obvious role, but beware getting too involved in technical nuances — users may throw you out**






3.3 Critical Success Factors for an E-Business BPR Project

-  Introduction
-  A vision of the reengineered environment
-  A reengineering methodology
-  Partnership participation from all levels of the organization, including executives and managers
-  Flexible, temporary teams
-  Continued quality improvement
-  Team training
-  “Systemic” thinking
-  Visible, active leadership

Introduction

-  **Several surveys indicate that approx 70-80% of BPR projects have failed, or have quietly disappeared without achieving results.**
-  **Some of the problem may be excessive “hype”**
-  **Some failures are inevitable, if the organization takes the “all-or-nothing” revolutionary approach articulated by Hammer et al**
-  **But many of the failures could have been predicted in advance**
-  **BPR projects have critical success factors; ignore them at your peril!**

A vision of the reengineered environment





-  People won't follow if they don't know where they're going (unless inspired by a charismatic leader).
-  People need a “model” of what is to be achieved
-  We must be able to tell them a “story” about the new world — broad-based, “horizontal” dissemination of this “story” is crucial, too (e-mail, video conferencing)
-  The “vision” must define the reengineered outcomes, processes, and behaviors that can be measured, reinforced, and practiced. It's hard to do this without a full modeling effort, but we need to do it quickly
-  Note that much of this also involves the (unspoken) concerns about the impact of the BPR on everyone's jobs, power, salaries, etc. (More on this later!)

A comment on “the vision thing”






“A process vision consisting of specific, measurable objectives and attributes of the future process state provides the necessary linkage between strategy and action. Unless such a vision is shared and understood by all the participants in a process innovation initiative—before redesign begins—the effort will all too easily slip from innovation to improvement.”

Tom Davenport, [Process Innovation: Reengineering Work through Information Technology](#)





A reengineering methodology

-  **There must be a step-by-step plan that addresses all areas of the organization impacted by the BPR project**
-  **It must compel people to work in teams for the long-term benefit of customers and the organization.**
-  **It must deliver clearly defined, executable and trackable implementation tactical plans.**
-  **And it must demonstrate, through the BPR work itself, the new behaviors expected in the new environment.**






Partnership participation

-  Participation is needed at all levels of the organization
-  ...including executives and managers
-  It's crucial to avoid the impression that "BPR is one of those crazy things IT is doing."
-  Ownership in the reengineering decisions is crucial to operational success
-  If BPR "problem" is pushed down to the level of bottom-level workers, it's usually impossible to solve the issues that cross departmental boundaries






Flexible, temporary teams

-  **Reengineering is not a life-time career!**
-  **Initial work should be spearheaded by temporary cross-organizational, cross-functional, and multi-level teams....**
-  **...who report to the BPR project executive sponsor.**
-  **Different teams will be needed at different stages in the BPR project.**




Continued quality improvement

-  This may be the focus anyway, if the process “innovation” turns out not to be radical, but merely incremental improvement
-  In any case, the “dramatic” BPR changes need to be followed up with continued improvements...
-  ...because customer and production needs will continue to evolve as time goes on.
-  Ongoing quality improvement should be done by the people in the reengineered work units, not by the BPR team.
-  Big challenge: avoiding “degeneration” back to old ways of doing things—the “crash diet” syndrome.








Team Training

-  **Because the BPR team is a new group of cross-functional people, they probably don't know how to work together**
-  **Team members typically don't know what their roles will be during and after the BPR effort.**
-  **Hence, miscommunication and political infighting is something to watch out for.**
-  **Team-building work and training exercises are often necessary, and more important than technical skills training**
-  **Technical skills: modeling techniques, simulation techniques, interviewing techniques, basic analysis techniques, etc, etc.**






Systems-thinking approach

-  Team should make its decisions based on numbers and facts—not politics, hunches, and intuition
-  Important to think “outside the box”—every system is a component of a bigger system
-  Common tendency to ignore “feedback” loops and focus instead on “cause-and-effect” situations

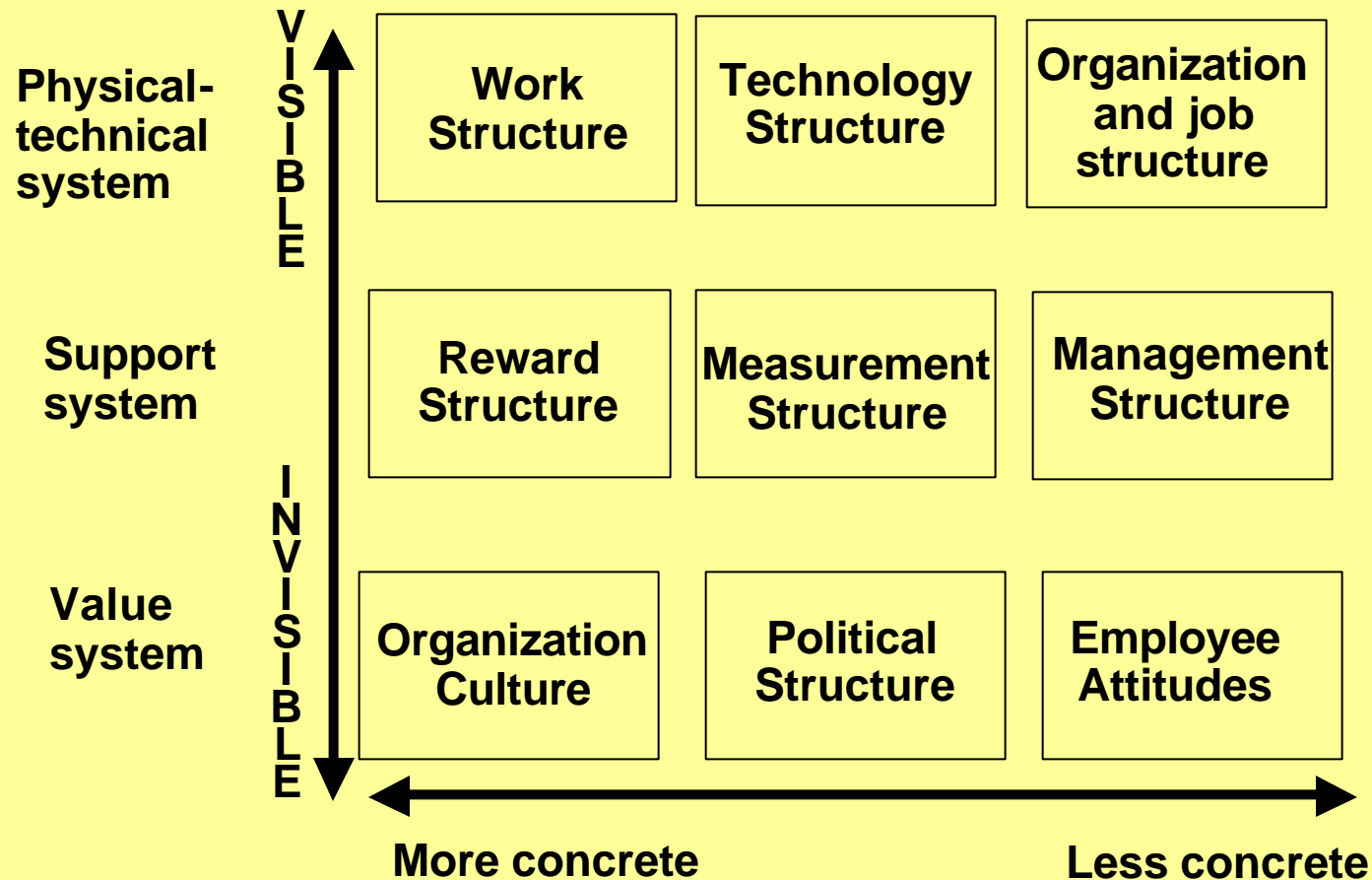
Visible, active leadership


-  **Common tendency for senior management to initiate a BPR project and then get distracted by ongoing crises.**
-  **Note that senior management is heavily involved in championing e-business initiatives. See “The Paradox of e-Business,” by Rob Austin, [Cutter Consortium Business Technology Trends and Impacts](#), Nov 2000**
 -  **IT initiatives have traditionally originated within IT department 61% of the time, senior mgmt 46% of time, business operating units 46%, marketing/sales 30%, finance 9%**
 -  **For e-business initiatives: senior management 30%, IT 22%, business operating units 20%, marketing/sales 9%, finance 2%**
-  **Can sometimes occur if too many BPR projects are initiated—US Defense Department has 230 BPR projects underway...**
-  **Management must focus on helping to articulate the problem and root causes—and not preach a predefined solution**
-  **Particular emphasis is needed on the “value system” and “support system” component of the BPR project (more on this later)**

3.4 A BPR Management Plan






-  **Identifying key areas of organizational impact**
-  **Planning for changes in the physical-technical system**
-  **Planning for changes in the support system**
-  **Planning for changes in the value system**
-  **A reengineering strategy**

An organizational impact model



 from "Business Reengineering," by Dorine C. Andrews and Susan K. Stalick, *American Programmer*, May 1992. See also their 1994 Prentice-Hall book, which discusses the same concepts in more depth.

Comments on this model








-  **IT people tend to concentrate on the top row, because it is the area they know best**
-  **Senior management may be seduced by the top row, because it is the most glamorous**
-  **Everyone focuses on the top row because it is the most visible**
-  **IT has difficulty dealing with bottom two rows because it is outside their political “turf”**
-  **...and because they often lack negotiation and assertiveness skills working with user depts**

Planning for changes in the physical technical system

Work
Structure

Technology
Structure

Organization
and job
structure








-  **Work structure involves reengineered workflow diagrams, and business procedural processes**
-  **Technology structure is where we see the glamorous new IT hardware/software stuff**
-  **Organization and job structure includes:**
 -  reporting and work group relationships
 -  accountabilities (who owns the process?)
 -  job content, skill, and knowledge requirements
 -  typically involves self-managed teams, reduced reporting levels, etc

Planning for changes in the physical technical system

**Work
Structure**

**Technology
Structure**

**Organization
and job
structure**








-  **This is where we ask the questions:**
 -  **What inputs and outputs are produced by this process?**
 -  **What steps must be carried out, and how must they be done?**
 -  **etc, etc, etc**
-  **Modeling techniques are useful here**
-  **Traditional techniques: DFDs, ERDs, etc**
-  **Most popular techniques today: use-cases, and UML-based OO analysis models**

Planning for changes in the physical technical system

Work
Structure

Technology
Structure

Organization
and job
structure








-  **Technology structure may introduce**
 -  client-server technology
 -  pen-based computing, voice-recognition, etc.
 -  PCs, workstations, GUI interfaces, etc.
 -  more on this in Section 8 of the seminar
-  **But there is a tendency for vendors to sell a “solution looking for a problem to solve”**
-  **Better to find appropriate technology structures after we have a good idea of the required work structure and processes**

Planning for changes in the physical technical system

Work
Structure

Technology
Structure

Organization
and job
structure






-  Changes in work structure and technology structure inevitably changes jobs & organization
-  Organization and job structure includes:
 -  reporting and work group relationships
 -  accountabilities (who owns the process?)
 -  job content, skill, and knowledge requirements
 -  typically involves self-managed teams, reduced reporting levels, etc
-  This stuff is “visible,” but it has an immediate impact on less visible areas of culture, etc.

Planning for changes in the support system

**Reward
Structure**

**Measurement
Structure**

**Management
Structure**






-  Rewards can motivate individuals and teams in a variety of ways—but changes are tricky, because they may lead to unanticipated consequences!
-  People generally do what they are paid to do and what they are recognized for
-  Politics or culture may make it difficult to make radical changes in reward structure to accompany radical changes in other parts of BPR project
-  Idealists may argue that it's impossible to develop a reward structure that will guarantee ideal process performance
-  But it's evident in many cases that any change in reward structure would be an improvement over current situation which virtually guarantees dysfunctional behavior

Planning for changes in the support system

Reward
Structure

Measurement
Structure

Management
Structure






-  Note that we want to measure outcomes and processes
-  Measurement should help reduce randomness and unpredictability in behavior and processes
-  Common problem in old process is that nobody measured anything
-  But failure to institute measurement structure often leads to disastrous results in new business processes
-  Example: the utility company that cleverly decided to remove the payment-due-date from the invoices it sent its customers...

Planning for changes in the support system

Reward
Structure

Measurement
Structure

Management
Structure





-  Management structure involves such things as whether people are included in business decision making
-  and availability and support for personal growth and development
-  and how the workers are treated (orders vs. consensus)
-  plus any other reinforcers of people's behavior on a day-to-day basis (can they telecommute? punch time cards?)
-  usually requires training and reinforcement for managers, too!

Planning for changes in the value system

Organization
Culture

Political
Structure

Employee
Attitudes






-  Organizational culture involves the rituals, myths, symbols, and language of “how we do things around here.”
-  Tasks are carried out in a certain way because they have become rituals over time—“I learned how to do it this way before your IT whippersnappers were even born!”
-  The older the culture, the more difficult it is to change
-  Dealing with this is often entirely beyond the ability of the IT professionals, especially because their culture may be perceived as being different from the rest of the organization

Changes in the value system, continued

Organization
Culture

**Political
Structure**

Employee
Attitudes

-  **Political structure includes formal and informal organization leaders**
-  **Centers of influence, unions, special interest groups, etc**
-  **Also informal relationships among key members**
-  **Leaders promulgate and reinforce organizational values and beliefs**
-  **An entrenched political structure may not wish to lose its power—hence the common cry to “obliterate, not automate” in BPR projects**

Changes in the value system, continued

Organization
Culture

Political
Structure

Employee
Attitudes

- ✍️ Aside from organizational culture and political structure, employees have their own “mental model” and beliefs that affect their attitude and behavior in the organization
- ✍️ Involves cultural characteristics like impatience, skepticism, openness, control, rigidity, or flexibility
- ✍️ This can be changed slowly, but requires active leadership, “walking the talk,” etc.
- ✍️ Or (in some cases), it can be changed by throwing out the “old guard,” and bringing in a fresh new generation of employees...

A Reengineering Strategy

Phase 1

**Create
the
framework**

Phase 2

**Conduct
pilot and/or
reengineering
projects**

Phase 3

**Institute
ongoing
quality
improvements**

Phase 1: create framework



Primary outcomes:

- a research “probe” into problems and opportunities
- a realistic “vision” of the reengineered future—critical!!!!
- an action plan



Often necessary to avoid 50 simultaneous, uncoordinated, helter-skelter BPR projects (example: American Express launched 285 reengineering projects in the 1992-93 time period, 100 were still active in mid-1994)



Should be accomplished quickly: 6-8 weeks normally



Beware the temptation to spend 5 years developing an all-encompassing “enterprise model” before work begins










Action plan should specify who will be involved in cross-organizational, multi-level teams—and who will be assigned to the reengineering projects

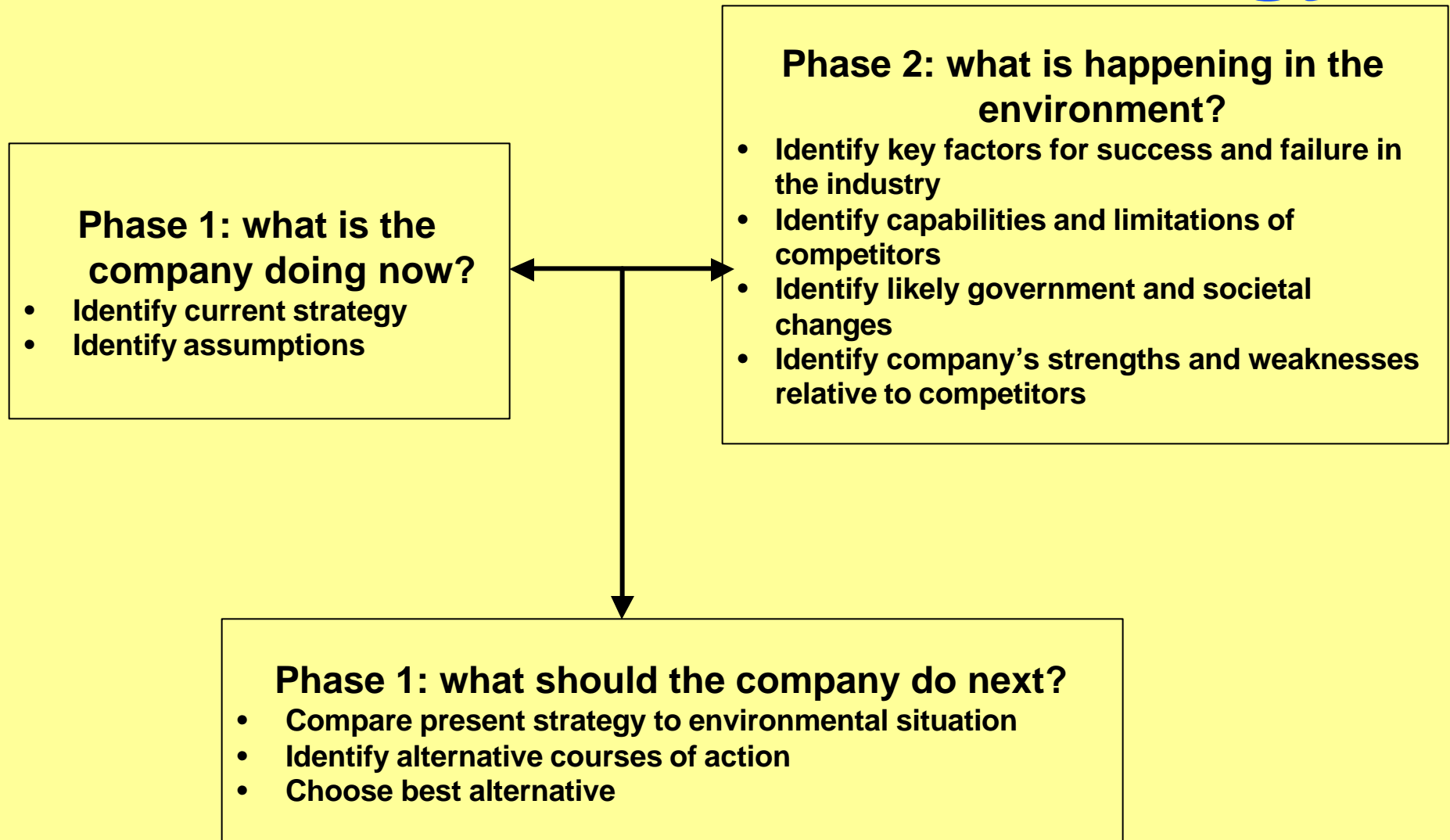


Creating framework tests for commitment from executive sponsors, and begins to identify changes needed in support system and value system of organization

Phase I, continued

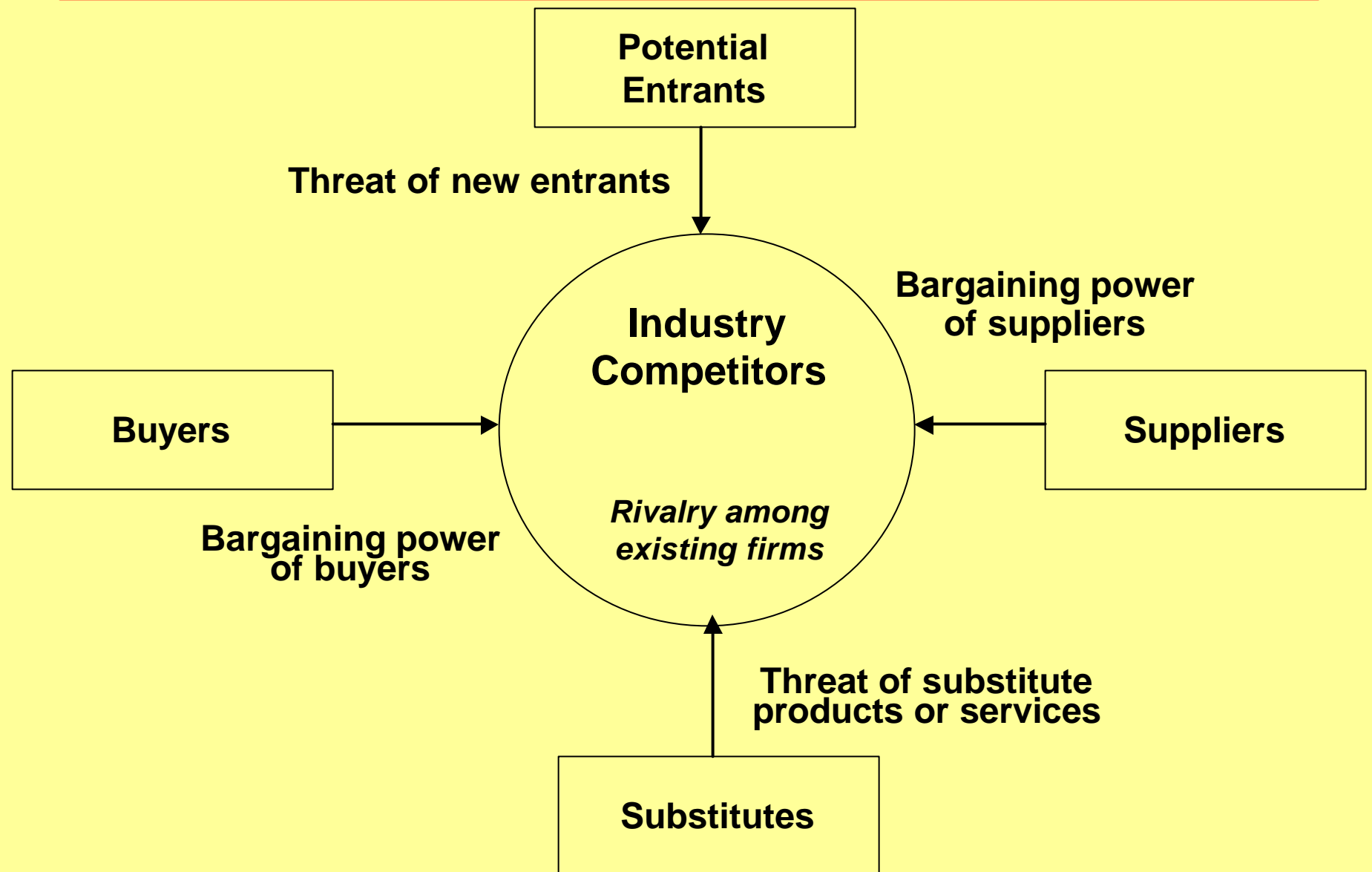
-  **Strategic planning has been a common practice for decades**
-  **But the “business cycle” has been speeding up so much in recent years that a “5-year plan” doesn’t seem as relevant any more**
-  **And strategic planning for e-business seems entirely different**
 -  **New questions about “what business are we in?”**
 -  **New questions about “core competence”**
 -  **Forces us to re-think basic assumptions, constraints**
-  **Yet many senior managers are demanding that their IT staff “get us on the Internet” before they have defined their e-business strategy**

What is a business strategy?












 See *Competitive Strategy*, by Michael E. Porter (Free Press, 1980)

Forces driving industry competition

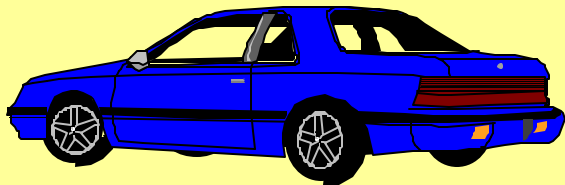


Comments on Business Strategies

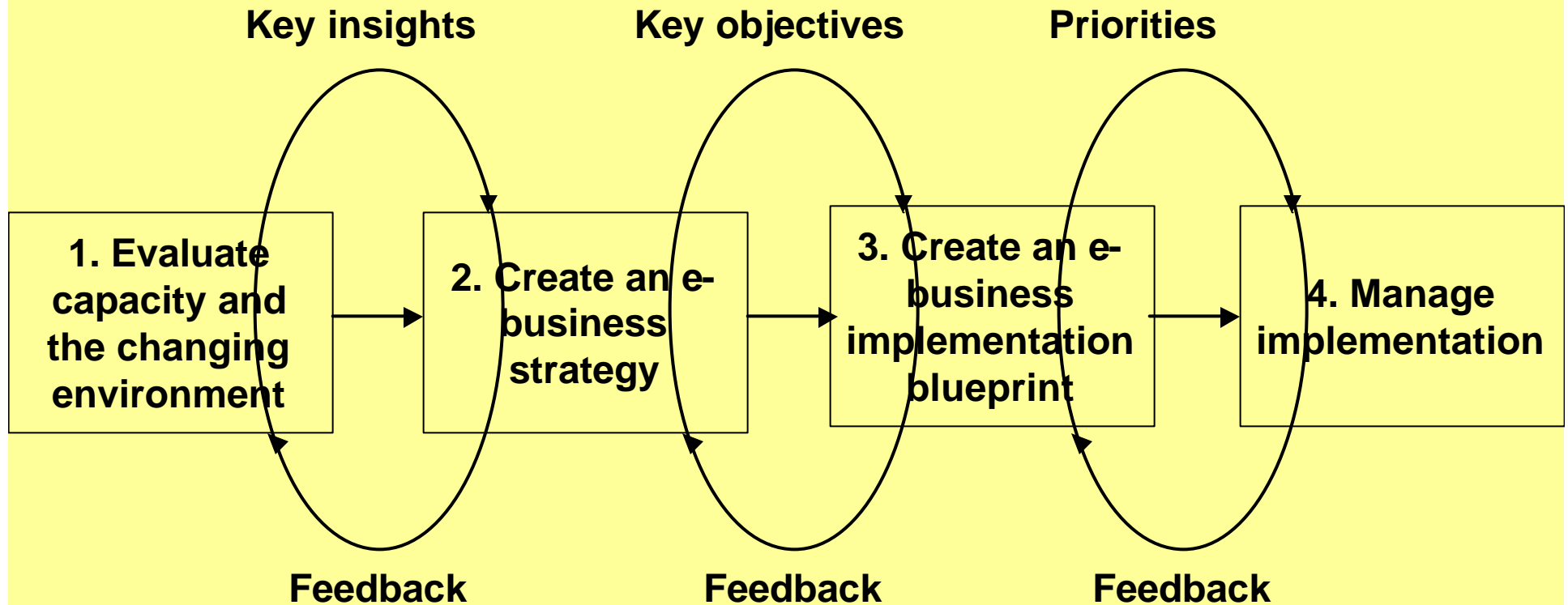
-  **Even though this is a familiar process, it doesn't mean everyone does it correctly**
 -  Western Union was offered patent rights to the telephone for \$10,000 but turned it down
 -  U.S. railroads didn't understand they were in the transportation business
 -  UK breweries didn't realize they were really in the real-estate business
-  **A recent survey indicates that 68% of organizations have a formal business strategy, 81% have a formal IT strategy; e-business strategy is included in 49% of business strategies, and 32% of IT strategies (see "Business-IT Strategies in Practice," by Chris Pickering, [Cutter Consortium Business-IT Strategies Advisory Service](#), Vol. 4, No. 5)**
-  **In a "stable" business environment, perhaps the process only needs to be repeated every 5 years**
-  **Implementation of the plan could usually be delegated to mid-level managers without much risk of trouble**
-  **As noted earlier, IT was often excluded from strategic planning activity**
-  **But all of that changes with e-business**

Key point about e-business

- ✍ **Key insight: Internet allows an e-business to separate “atoms” from “bits” and distribute them separately**
- ✍ **Most traditional businesses have embedded information about a product into the product itself**
- ✍ **Competitors can often “grab” the (profitable) information without being stuck with the (less profitable) product**
- ✍ **Classic example: music “published” on CD/tape**
- ✍ **Similarly, information about customers, suppliers may be more valuable than “owning” them.**








The new e-business strategy









 **From *e-Business: Roadmap for Success*, by Ravi Kalakota and Marcia Robinson**

Comments on the new strategy

-  **Since things are moving so quickly, all of these activities need to be going on in parallel.**
-  **Rapid changes also require more feedback and iteration than ever before.**
-  **Implementation issues cannot be ignored by senior management; hence it is part of the plan**
-  **E-business often requires difficult re-engineering of IT architecture, hence needs senior management involvement**
-  **“Time-to-market” pressure of e-business requires difficult business prioritization, hence senior management involvement**

Basic e-business strategies

-  **Some companies may give up their traditional business of selling “atoms+bits” and sell only “atoms” or “bits”**
-  **But most traditional companies will be reluctant to go so far; but they still need to separate the processes and streamline them**
-  **Look for new “value propositions” to offer customers, rather than refining/optimizing existing value propositions**
 -  **Because traditional companies typically focus on “economies of scale” to optimize manufacturing/production of “atoms”**
 -  **They tend to downplay the “bits” aspect, leaving themselves vulnerable to competition from upstarts.**
-  **Recognize that Internet makes customers more powerful, so focus on customer value propositions first, then figure out how to supply that value**

Three general strategies



Customer-focused

- If product is a commodity, then emphasize the relationship with the customer
- Optimize the buying experience
- Foster communities of users
 - visiting Web sites indicates that most companies are “clueless” about how to do this
 - See The [The Cluetrain Manifesto](#) for good discussion of this point.



Operations-focused








- If product is a commodity, then become the low-cost, most trouble-free supplier of the product
- Use Internet to optimize the supply chain
- Disintermediate, and/or “re-intermediate”









Product-focused

- If product is high-tech, then emphasize new versions of product, optimize upgrades, etc.
- Maximize flexibility







Strategy implementation issues

-  **Fundamental question: how can you radically change the culture to accommodate the e-business view?**
-  **How can you develop new products/services in “Internet time” when you haven’t done it before?**
-  **How can you change core business processes without also modifying IT processes and infrastructure?**
-  **How can you interface new IT architectures with existing legacy infrastructures?**
-  **Basic answers:**
 -  **Some organizations will find that they can only accomplish this by “spinning off” independent subsidiaries, or acquiring small companies**
 -  **Some organizations will not be able to do this at all — e-business will be a “Darwinian event” in the business world.**







E-Business Strategy References

-  Michael Porter, *Competitive Advantage* (Free Press, 1980).
-  Ravi Kalakota and Marcia Robinson, *e-Business: Roadmap for Success* (Addison-Wesley, 1999)
-  Philip Evans and Thomas S. Wurster, [Blown to Bits](#) (Harvard Business School Press, 2000)
-  Patricia Seybold, with Ronni T. Marshak, *Customers.com: How to Create a Profitable Business Strategy for the Internet and Beyond* (Times Business/Random House, 1998)
-  Carl Shapiro and Hal R. Varian, *Information Rules: A Strategic Guide to the Network Economy* (Harvard Business School Press, 1999)
-  Keyur Patel and Mary Pat McCarthy, *Digital Transformation: The Essentials of e-Business Leadership* (McGraw-Hill, 2000)

Analyzing Current Situation























-  **Not all systems are good candidates for BPR — need to focus efforts on high-priority areas**
-  **Discuss following questions in one-on-one interviews, or “focus groups” of 4-8 people**
-  **...or through on-site observation of the process**
-  **or by talking to vendors and customers!!!! They have an entirely different perspective**
-  **Collect hard data/metrics wherever possible**
-  **Also consider using these questions to benchmark competitors and “best-in-class” performers outside your industry who have similar processes (e.g., customer service, order entry, etc.)**

Analyzing Current Situation, cont'd

-  Questions on following pages sometimes difficult to answer if you don't have a process model...
-  ...but they can help focus attention on the areas where a process model is most needed
-  Key question: to what degree do the issues on the following pages influence tenfold improvement in the core processes?
-  Often useful to have facilitated JAD sessions here to encourage “neutral” gathering of issues
 -  low-tech approach: bulletin boards, “Post-It” notes, etc.
 -  high-tech approach: anonymous terminals connected to an electronic display board for collection, prioritization of issues











1. What prevents you, in the business processes you perform, from satisfying your customers and creating quality products and services?

Probe: *How much/many? What frequency? What trends?*

- | | |
|--|--|
|  Time delays |  Duplication of work |
|  Handoffs |  Lack of standards |
|  Transaction error |  Documentation errors |
|  Lack of controls |  Transaction volatility |
|  Rigid procedures |  Fragmentation of work |
|  Exception processing |  Inconsistent inputs to work |
|  Facility problems |  Inaccurate inputs to work |
|  Paperwork problems |  Unclear work outputs |
|  Approval layers |  Policy problems |
|  Review cycles |  Procedure problems |
|  <i>Resource constraints</i> |  Content complexities |











2. What does technology or the lack of it do to enhance or inhibit effective process performance?

Probe: *With customers? With internal people?*

-  **Communication**
-  **Information access**
-  **Data creation, updating, deleting**
-  **Decision support**
-  **Transaction processing**
-  **Outcome production**
-  **Information timeliness**
-  **Information availability**
-  **Performance monitoring**
-  **Work flow handling**






3. What does the organization structure do to enhance or inhibit effective process performance?

Probe: How much/many? What frequency? What trends?

-  Job position structure
-  Job reporting relationships
-  Job content
-  Job skill/knowledge requirements
-  Job accountabilities
-  Job complexity
-  Organization structure
-  Job groupings
-  Work relationships
-  Organization type (military, consensus, team-based,etc)

4. What do the reward structures (financial, nonfinancial, formal, informal) do to enhance or inhibit effective process performance?

Probe: *How often? What kind?*

-  **Alignment/nonalignment with process performance objectives**
-  **Consistency of application**
-  **Clarity of definition/understandability**
-  **Relationship with actual process performance**
-  **Discrepancies**

5. What do the measurement structures or lack of them do to enhance or inhibit process performance?

Probe: *How much/many? What frequency? What trends?*

 **Customer satisfaction**

 **Quality of process outcomes**

 **Dates met**

 **Consistency**

 **Appropriateness**

 **Accuracy**

 **Completeness**

 **Requested changes**

 **Volumes produced**

 **Process performance**

 **Efficiency**














 **Cost**

 **Profitability**

 **Accuracy and errors**






6. What do management methods or lack of them do to enhance or inhibit process performance?

Probe: How often? What trends? What consistency across different managers?

-  **Leadership capabilities**
-  **Leadership style**
-  **Control of decision-making**
-  **Management style (hands-on, coaching, etc)**
-  **Performance development support**
-  **Performance management support**
-  **Qualification of managers**
-  **Managerial experience**
-  **Decision style**
-  **Predictability of rule enforcement**
-  **Degree of subordinate involvement in decision making**
-  **Praise from managers**
-  **Punishment from managers**








7. What does the culture do to enhance or inhibit process performance?

Probe: *What kind? To what degree of severity?*

-  **Rituals, symbols, and myths**
-  **Language**
-  **Attention/focus (internal, external, solution, problem, etc)**
-  **What is important (people, customers, things, tools, etc)**
-  **Position in industry**
-  **Position to customers**

8. What does the political power within the organization do to enhance or inhibit process performance?













Probe: *Key players? How often? To what degree?*

-  **Coercive/punishment power used**
-  **Influencers**
-  **Legitimate power sources**
-  **Personal power sources**
-  **Power styles (confrontational, subversive, etc)**
-  **Empowerment of subordinates**
-  **Focus of power (for organization, for personal use, etc)**










9. What do the belief systems of individuals do to enhance or inhibit process performance?

Probe: *Key individuals? How extreme?*

About...












-  **Customers**
-  **How things work (get done)**
-  **Change**
-  **Accountability**
-  **Competency of others**
-  **Trust in leaders**
-  **Products and services**
-  **Organization mission**
-  **Organization culture**
-  **Themselves**
-  **Work environment**
-  **Ability to influence others**

Phase 2: Pilots and/or Reengineering projects








-  These will normally go on for 12-18 months (or more)
-  Often necessary to have a central coordination/support team that addresses “generic” BPR issues and problems
-  Problems to watch out for:
 -  Becoming overwhelmed with the possibilities of “what could be”
 -  Impatience: often important to have 6/12/18 month checkpoints to see progress in fixing a problem that took 20 years to develop
 -  The temptation to turn the BPR project over to a few “experts” or outside consultants; involved parties sometimes feel “too busy” to participate
 -  Paralysis of perfection—need to focus on workable solutions, with the possibility of ongoing incremental improvements later
 -  Pressure from political powers—especially from those who feel they are not part of the BPR process, and who feel threatened by it.
 -  Financial support—especially if the BPR project extends from one budget cycle into the next one.

Phase 3:
















Ongoing Quality Improvement

-  Reengineered concepts and processes need to be institutionalized to avoid returning to the “old ways”
-  Continued, incremental improvement must be assured through ongoing quality improvement techniques
-  Quality improvement process should incorporate:
 -  measurement of processes and results to find data for trend analysis
 -  analysis: reviewing the data to spot crises before they occur
 -  solution development—using creative problem-solving techniques
 -  experimentation, allowing for prototypes, test-beds, to see what works
 -  implementation: the people who found the problem can implement solutions, in most cases
 -  reporting: to review previously installed solutions for progress
 -  raising the standards—revising performance and quality objectives
 -  recognition—identifying individuals and teams who have helped achieve quality and performance improvements






4. PROJECT NEGOTIATIONS

-  **Managing project definition at the beginning of the project**
-  **Using project definition to manage requirements creep**
-  **Estimating techniques**
-  **Tools for assisting estimation process**
-  **Tradeoffs between schedule, budget, staff, quality**
-  **Tools for rational negotiation**
-  **What to do when rational communications are impossible**







4.1 Managing Project Definition: What does “success” mean?

-  Many projects succeed or fail at the very beginning, before any technical work is done.
-  Fundamental requirement: identifying who has the right to declare “success” — owner, shareholder, etc, etc.
-  Typical definitions of “success”
 -  finishing on time
 -  staying within budget
 -  delivering the required functionality
 -  providing “good enough” level of quality
 -  getting the next round of VC funding, or launching the IPO
-  Key indicators of a doomed project:
 -  Nobody can articulate what “success” means for this project...
 -  ...or it’s such a vague definition that it will be hard to demonstrate success
 -  Key stakeholders cannot reach agreement on definition of success
-  The combination of these constraints may prove impossible to achieve — so the pragmatic aspect of success often depends on agreement as to which areas can be compromised or satisfied.
-  Biggest risk: lack of realistic triage at beginning of project
-  For more details, see my articles, “[Spelling Success](#),” *Computerworld*, Feb 19, 2001; and “[Long-Term Thinking](#),” *Computerworld*, Oct 26, 2000; and “[The Value of Triage](#),” *Computerworld* , March 20, 2000








4.2 Using Project Definition to Manage Requirements Creep

-  Typical behavior in projects: new requirements are added at the rate of 1% per month
-  Requirements “creep” and requirements “churn” are a major element of project management risk — particularly in e-business projects that often involve fundamental changes to business processes.
-  But if you don’t have a formal document describing the requirements, it’s hard to identify creep or churn.
-  Assuming that you do have such a document, you need to use it to negotiate schedule/budget/staff modifications if the requirements change or increase.
-  Biggest risk of all: an ambiguous spec is usually a sign of unresolved conflict between diverse political camps in the user community. Related risk: techies assume that it’s their fault they can’t understand ambiguous spec. For more about this, see my Nov 2000 *Computerworld* article, “[How to Help Users Write Good Requirements Statements](#)”

4.3 Estimating Techniques














-  **Fundamental truth: to estimate a project you need metrics from previous projects. Most IT organizations have almost no metrics about previous e-business projects.**
-  **Thus, most of what's described as "estimating" is either "guessing" or "negotiating" (see "Metrics and the Seven Elements of Negotiation," by Michael Mah, [IT Metrics Strategies](#), April 2001)**
-  **Political reality: estimates are produced by people with little prior estimating experience, and who have a vested interest in believing their optimistic predictions**
-  **A radical suggestion: create a separate estimating group whose work is judged and rewarded by the accuracy of its estimates, not the political acceptability of estimates**
-  **For a new approach to estimating, see "critical chain" approach, which "pools" safety estimates across several tasks within a project. See Eliyahu M. Goldratt, [Critical Chain](#) (North River Press, 1997), and "Software Critical Chain Project Management: Do Silver Bullets Exist for Schedule Reduction?", by Richard E. Zultner, [Cutter Consortium Business-IT Advisory Service](#), Executive Report, Vol. 3, No. 10**
-  **For complex projects, get a commercial estimating tool**

4.3, cont'd: Metric-Based Scheduling and management








-  **Cost and schedule estimates should be based on empirical data**
-  **Metric-based planning requires early calculation of size, projection of costs and schedules from empirical patterns...**
-  **...and tracking of project status through the use of captured result metrics.**
-  **The important issue here is to identify problems early**
-  **Metrics are the yardstick for measuring progress against your baseline plan, and thus become your warning indicator for further inquiry and action.**
-  **It's important to determine whether an indicated schedule delay is a problem with the people carrying out the effort, or a plan that was too aggressive to begin with.**
-  **Don't punish the project team if they don't meet unreasonable cost and schedule estimates (if they believe that's their fate, they may give up and practicing a "work to rule" approach).**

Metric-Based Scheduling/Estimating






“Best Practice” Essentials

-  Estimate cost and schedule using data from completed projects of similar size and objectives
-  Compare with cost model estimates
-  Plan short-duration tasks with measurable products (preferably binary in nature)
-  Review key metrics at frequent intervals throughout the project:
 -  Earned value vs. “actual expended”
 -  Cost to complete (including estimate for unresolved risk) vs. “planned at completion”
 -  Schedule to completion vs. planned schedule
 -  cost performance index
 -  to-complete performance index
-  Manage defect closure time (see “defect tracking against quality targets” below)
-  Don’t hide problems by constantly re-baselining
-  Report earned value, etc. against original baseline
-  Track other control panel metrics





Metric-Based Scheduling/Estimating Status Checks (from the Airlie Council)

-  Does the plan identify progress measures to permit rate charting and tracking?
-  Are inspection coverage and removal rates tracked for the entire product and for each component?
-  Are project estimates continuously refined as the project proceeds?
-  Is a project feedback loop established between project measures and updated schedules?
-  Is there a process for capturing the primitive data necessary to calculate the earned value?
-  Are productivity levels and schedule deadlines evaluated against past performance and reflected in the risk assessment?
-  Is the planned versus actual cost and planned versus actual schedule monitored?

4.4 Tools for Estimating

-  **KnowledgePlan**, from Software Productivity Research
-  **SLIM**, from Quantitative Software Management
-  **ESTIMACS**, from Computer Associates
-  **COCOMO-2**, available from several commercial vendors
(See CoStar from SoftStar Systems)
-  **OnYourMarkPro**, from **Omni-Vista** (caveat emptor: I'm on the Board of Technical Advisors at this company)

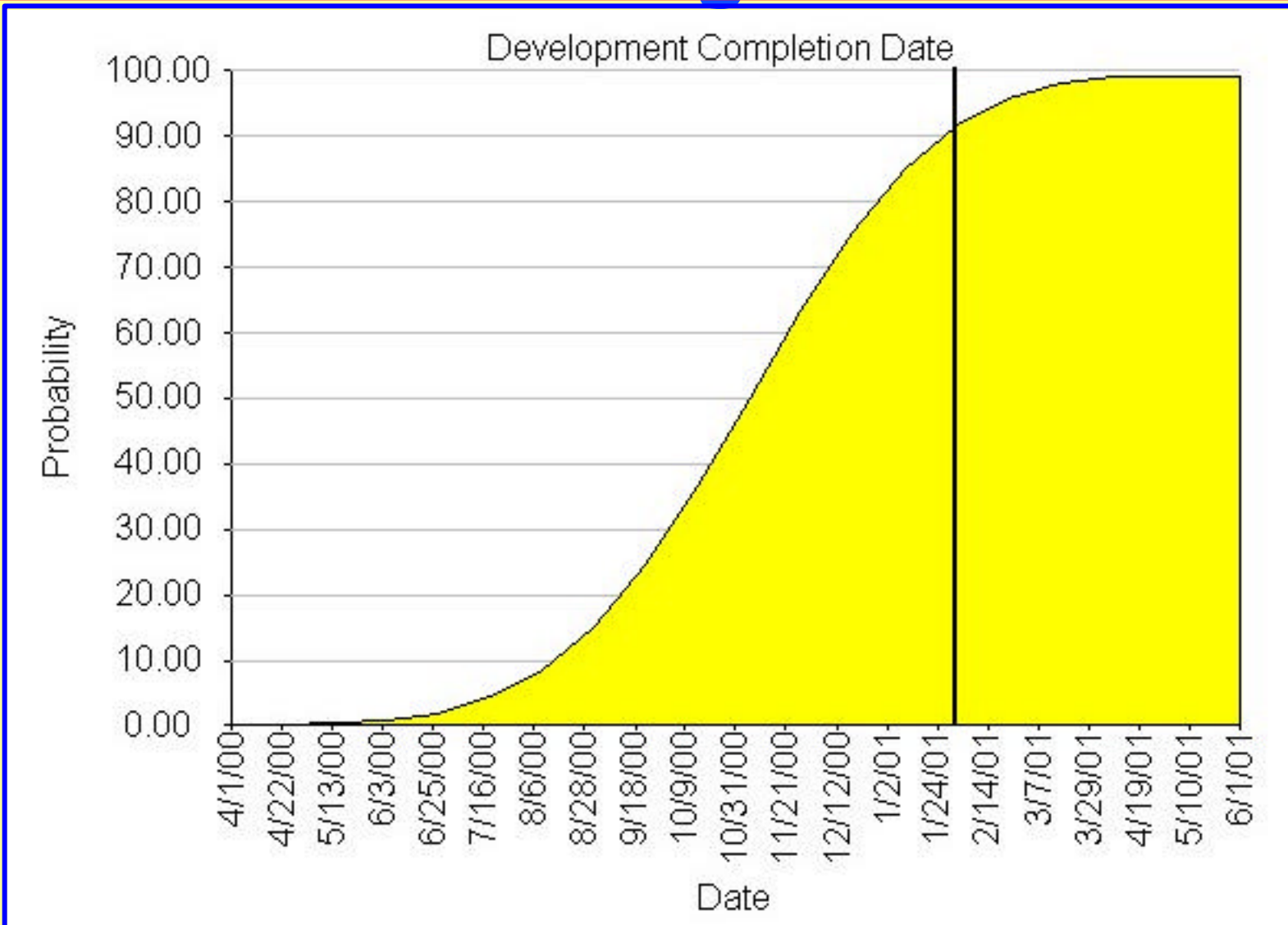
4.4, cont'd: Additional strategies for estimating

-  System dynamics models, e.g., Tarek Abdel-Hamid's model in [iThink](#)
-  Copies of [The Mythical Man-Month](#) for all concerned
-  ...or copies of Tom DeMarco's [The Deadline](#) for those who prefer a less serious book.
-  Time-boxing to see how feasible/infeasible the project constraints really are




4.5 Tradeoffs between schedule, budget, functionality, staff, quality

- ✍ Key point: it's not a linear tradeoff — see Fred Brooks, [*The Mythical Man-Month*](#) (Addison-Wesley, 1995)
- ✍ Relationship is a non-linear, third-order polynomial relationship — see Larry Putnam and Ware Myers, [*Measures for Excellence: Reliable Software on Time, Within Budget*](#) (Prentice-Hall, 1992)
- ✍ Biggest risk: tradeoffs are usually negotiated, under pressure, late in the project schedule — without accepting the non-linear tradeoffs...
- ✍ ...and without accepting the reality that much of the partially-finished work will be lost forever
- ✍ To negotiate tradeoffs rationally, you need to have one of the estimating packages mentioned earlier
- ✍ For more details, see “Internet-Speed Deadline Management: Negotiating the Three-Headed Dragon,” by Michael Mah, [*IT Metrics Strategies*](#), May 2000




Typical trade-off chart from estimating tools














4.5, cont'd: “Rules of Thumb” for Estimating Tradeoffs

-  Presumes that you have a “rational” estimate of optimal schedule, budget, resources, etc.
-  A change of up to 10% in one dimension can usually be accommodated with a proportional change in another dimension — though not necessarily at the end of the project!
-  Above 10%: use the “square law”








4.6 Project Negotiations

-  **Beware the temptation to give up... e.g.,**
-  **“We have no idea how long this project will really take, and it doesn’t matter, since they’ve already told us the deadline...**
-  **...so we’ll just work 7 days a week, 24 hours a day, until we drop from exhaustion. They can whip us and beat us, but we can’t do any more than that...”**








4.6, cont'd Negotiating games

-  Doubling and add some...
-  Reverse doubling
-  Guess the Number I'm Thinking of...
-  Double Dummy Spit
-  The X-Plus Game
-  Spanish Inquisition
-  Low Bid
-  Gotcha — throwing good money after bad
-  Chinese Water Torture
-  Smoke and Mirrors/Blinding with Science
 -  thanks to Rob Thomsett, "Double Dummy Spit, and Other Estimating Games," American Programmer (now [Cutter IT Journal](#)), June 1996

4.6 Negotiating strategies

-  Don't get tricked into making an “instant estimate” — ask for time to think about (a week, a day, even an hour)
-  State the estimate in terms of confidence levels, or \pm ranges, etc.
-  Jim McCarthy (formerly of Microsoft, author of [*Dynamics of Software Development*](#)): make the customer, or other members of the organization, share some of the uncertainty.
-  Project manager: “I don't know precisely when we'll finish — but I'm more likely to be able to figure it out than anyone else in the organization. I promise that as soon as I have a more precise estimate, I'll tell you right away.”
-  Do some reading and research to become better at this area, e.g.:
 -  [*Bargaining for Advantage: Negotiating Strategies for Reasonable People*](#), by G. Richard Shell (reissue edition, Penguin Books, June 2000)
 -  [*Getting Past No: Negotiating Your Way from Confrontation to Cooperation*](#), by William Ury (Bantam Doubleday Dell, 1993)

4.7 What to do when rational negotiation breaks down

-  Quit (the project or the company)
-  Appeal to a higher authority
-  Go see the movie [Gladiator](#), and learn to say, like Russell Crowe, “We who are about to die salute you!”
-  Decide which “rules” you’re going to break in order to achieve an “irrational” set of schedule/resource demands that have been imposed upon you.
-  Redefine the project as a kamikaze, suicide, etc., and make sure entire project team knows it.
-  Key point: project leader has to believe in the possibility of achieving project goals
-  ...and must be able to convince team members without “conning” them

A comment from Napoleon










- ✍ “It follows that any commander in chief who undertakes to carry out a plan which he considers defective is at fault; he must put forth his reasons, insist on the plan being changed, and finally tender his resignation rather than be the instrument of his army’s downfall.”
- ✍ Napoleon, [Military Maxims and Thoughts](#); see also [The Military Maxims of Napoleon](#), by William E. Cairnes (De Capo Press, 1995)

4.7, cont'd:

Quitting and the “social contract”

- ✍ Traditional corporate culture used to be based on “a job for life” — like marriage, till death do us part...
- ✍ Which meant that we were expected to put up with a lot of grief in high-pressure projects
- ✍ But many employers have demonstrated that the “social contract” is no longer valid (especially when an economic recession occurs; see my Jan 2001 Computerworld article, “[Managing Projects in Times of Recession](#)”)
- ✍ If the employer threatens to fire you if the e-business project fails, then you should be equally cold-blooded if you’re given impossible constraints for the project




5. PEOPLEWARE ISSUES

-  **Hiring and staffing issues**
-  **Recruiting**
-  **Identifying loyalty and commitment issues**
-  **Importance of communicating project priorities**
-  **Team-building issues**
-  **The manager's role in the team**
-  **Establishing proper peopleware "culture"**
-  **Project training and management "flight simulators"**
-  **The role of "end-user" manager in leading projects**











5.1 Hiring and Staffing Issues

- ✍ Strategy #1: hire superstars and turn them loose
- ✍ Strategy #2: insist on a well-honed “mission impossible” team that has worked together before
- ✍ Strategy #3: choose mere mortals, but make sure they know the risks and rewards associated with the project
- ✍ Strategy #4: take whoever you’re given and convert them into a mission-impossible team (e.g., the [Dirty Dozen](#) approach)

5.1 Hiring and Staffing, cont'd

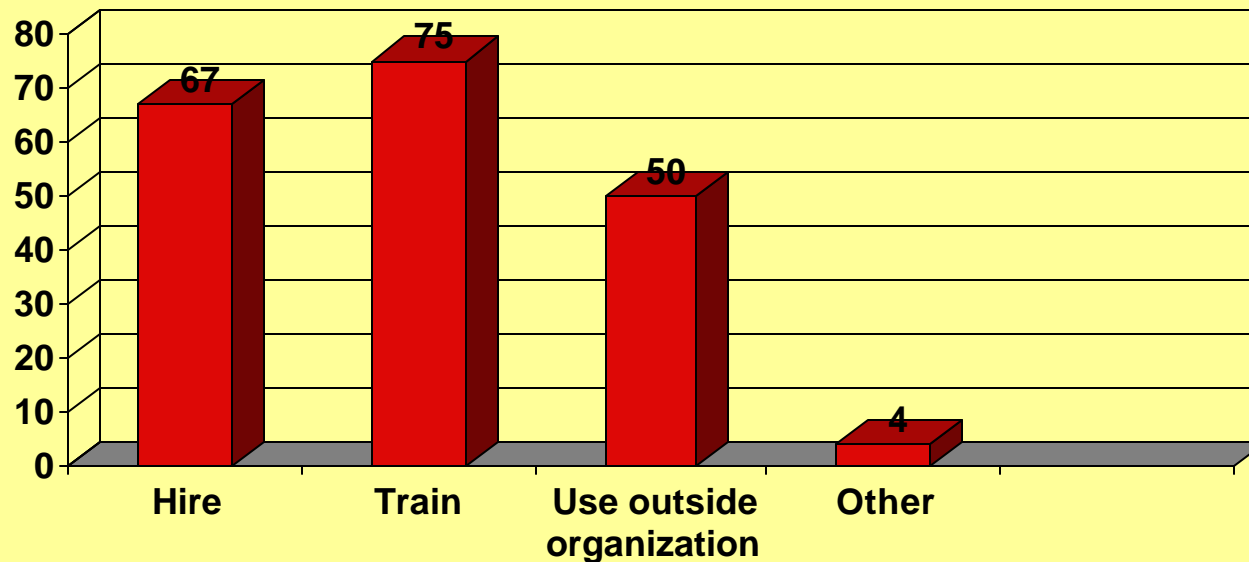
-  Risk increases substantially if project manager cannot choose his/her team members...
-  ...or at least veto the assignment of totally unacceptable morons to his/her team.
-  Crucial to avoid losing people during the project; highly desirable not to add new people during project

5.2 The Importance of Recruiting






-  Don't let HR department dominate.
-  Offer bonuses to existing staff members who can recommend new people
-  Develop a good relationship with local universities who may be a “spawning ground” for new people
-  Send top IT managers to university campus to recruit new graduates
-  “Eat your own dog-food”: develop an e-business process for recruiting (which may require some BPR work)
 -  Advertise job openings on the Web
 -  Allow prospective people to send e-mail inquiries via the Web
 -  Collect resumes on the Web
 -  Interview (with video hookup) via the Web
-  *Recruit continuously*

5.2 The Importance of Recruiting















- ✍ Recent study found that only 30% of IT organizations have enough developers for their projects (see Paul Harmon, “Finding Developers for E-Commerce Projects,” [Cutter Consortium](#) Distributed Computing Architecture/e-Business Advisory Service, April 2000)
- ✍ How do organizations acquire additional people for their e-business projects?









Teaching IT Managers How To Interview

-  **Don't assume technical manager know how to do this!!**
-  **Observe an interviewing session**
-  **Many managers use their interview to discuss themselves**
-  **Train managers who need interviewing skills**
-  **Ask candidates to bring a portfolio of their work.**





A Metaphor

-  **Circus Manager:** How long have you been juggling?
 **Candidate:** Oh, about six years.
-  **Manager:** Can you handle three balls, four balls, and five balls?
 **Candidate:** Yes, yes, and yes.
-  **Manager:** Do you work with flaming objects?
 **Candidate:** Sure
-  **Manager:** ...knives, axes, open cigar boxes, floppy hats?
 **Candidate:** I can juggle anything.
-  **Manager:** Do you have a line of funny patter that goes with your juggling?
 **Candidate:** It's hilarious.
-  **Manager:** Well, that sounds fine. I guess you're hired.
-  **Candidate:** Ummm ... Don't you want to see me juggle?
 **Manager:** Gee, I never thought of that.
-  **Tom DeMarco and Tim Lister, [Peopleware](#)**






The Audition Approach

-  **Complements interviews, reference checks, etc.**
-  **Ask candidate to give oral presentation to a group of technicians, managers, and non-technical people**
-  **Good presentation topic: the candidate's last project.**
-  **Alternative: ask them to write some code, develop a test case, design a database schema, etc.**
-  **When finished, get a consensus from the group: yes, no, or "maybe".**
-  **Don't overlook the social impact: passing the audition is a "certification" process; failing the audition improves the morale of the existing staff members.**




Audition Checklist

-  **Energy level**
-  **Ability to communicate in whole sentences**
-  **Ability to answer questions**
-  **A sense of excitement/passion about his/her work**




4.3 Identifying Loyalty and Commitment

-  **Ideal case: loyalty to project comes before all else.**
-  **Often associated with young, unmarried techno-nerds, with no life outside the office**
-  **May also depend on charisma of project leader**
-  **Also depends on length of project: total devotion is feasible for 3-6 month e-business project, but not = 24 months**
-  **Crucial that everyone knows what the loyalty issues are — e.g., “If my spouse threatens to divorce me, then I’ll have to quit this project.”**






Motivating Software People

-  Normal motivation techniques
-  Breakthrough Performances
-  “Voluntary” Overtime




A thought about motivation

-  “There is nothing more discouraging to any worker than the sense that his own motivation is inadequate and has to be ‘supplemented’ by that of the boss...”
-  You seldom need to take Draconian measures to keep your people working; most of them love their work.”
-  Tom DeMarco and Tim Lister, [Peopleware](#)

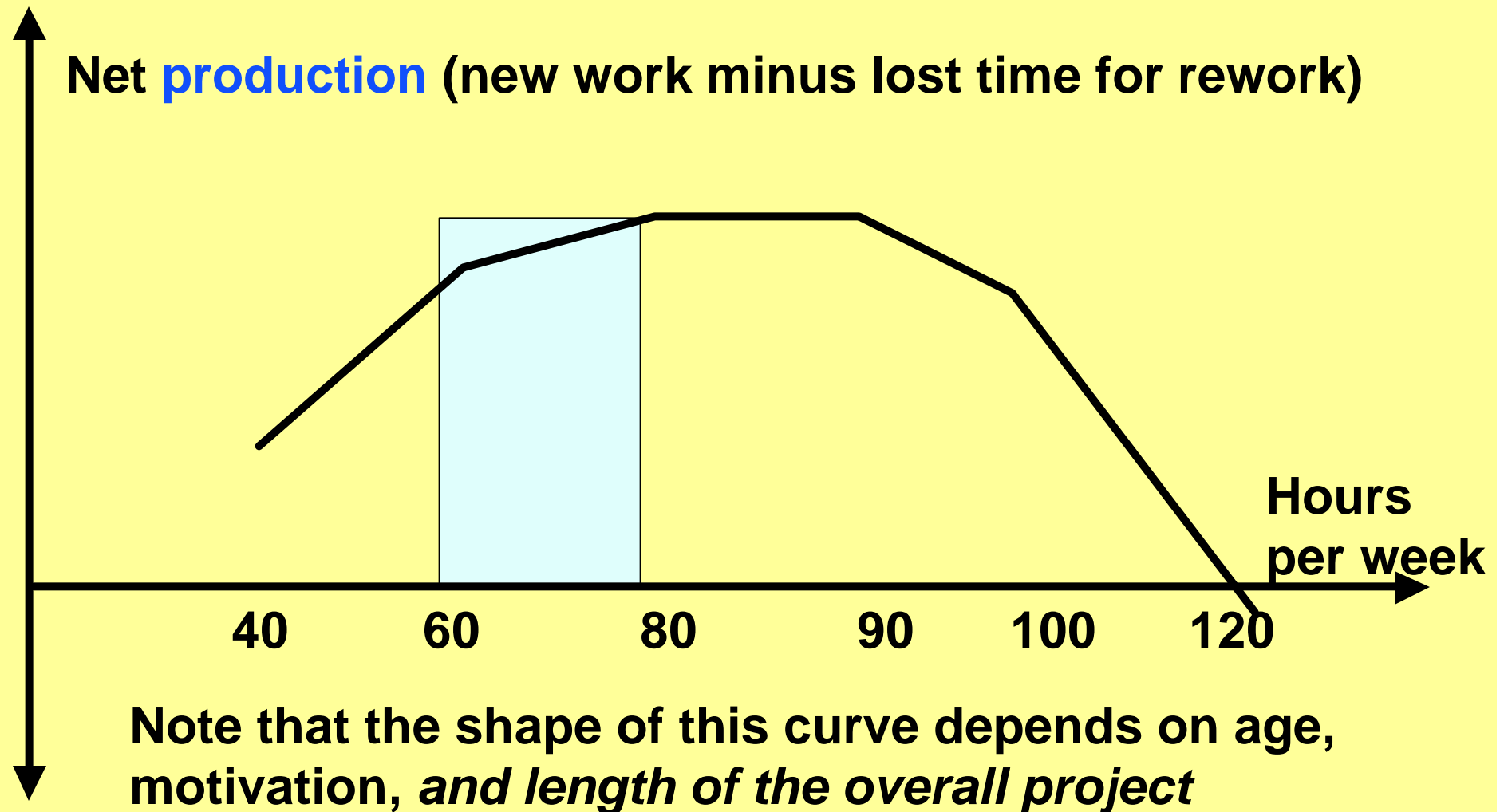
“Voluntary” overtime

-  An inevitable part of many software projects, especially high-intensity e-business projects
-  Worst mistake is not recording it, so that it appears to be “free” and “infinite”
-  Typically easier to accomplish if project team is young, unmarried, and fanatical
-  Also easier to accomplish if overall duration of project is 3-6 months.
-  Beware the insidious effects of long-term heavy overtime on projects of 18-24 months, etc.

John Boddie on overtime

-  “Individuals have different metabolisms. Some are night people, others work better in the early morning. Irrespective of type, nobody’s health is going to be ruined by working ten-hour days. **Once the project gets rolling, you should expect members to be putting in at least 60 hours per week.** If they’re not, check first to see if there’s something in the way the project is organized that’s frustrating them.
-  “**The project leader must expect to put in as many hours as possible.** This is done for two reasons. First, he must provide an example. You cannot expect people to work overtime if you’re not doing it yourself. Overtime must be led. Second, he must be there to answer questions, cut through red tape, and fix problems that come up during odd hours.
-  John Boddie, *Crunch Mode*, page 124 (Prentice Hall, 1987)

Overtime and net productivity



The Impact of Money




“Money, benefits, comfort, and so on are “hygiene” factors—they create dissatisfaction if they're absent, but they don't make people feel good about their jobs and give them the needed internal generator. What does produce the generator are recognition of achievement, pride in doing a good job, more responsibility, advancement, and personal growth. The secret is job enrichment.”

Frederick Herzberg,





“One More Time: How Do You Motivate Employees?”

Harvard Business Review, Sep-Oct 1987






More about money

-  A 10% salary increase is typically *much* more important to a junior person than someone with 10+ years of experience — but today's junior-level people have high expectations!
-  Many Silicon-Valley software companies pay salaries 10-20% *below* average, but compensate with stock options and other long-term benefits (dot-com collapse is making some companies re-think this strategy!)
-  But if *enormous* income is not a possibility, then salary/compensation is more of a “hygiene” factor than a positive incentive.




More about money

-  If you're going to pay overtime or bonuses, base it on work carried out; beware dangling the carrot of "successful completion" in front of the programmers, because they may not be able to control success of the project
-  Various other quasi-monetary and non-monetary rewards may sometimes be just as valuable as overtime pay (e.g., ability to keep one of the development machines at the end of the project)
-  Project manager needs to have a budget item for late-night meal, Friday-night beer & pizza, etc. Don't nickel-dime the programmers in this area! *Microsoft refers to this as a "morale" budget*
-  Another example: concierge services such as [Circles](#), [LesConcierge](#), [Peapod](#), Streamline, [Webvan Group](#) for chores that programmers don't have time to do themselves. (Alas, some companies in this area have gone bankrupt!)


Breakthrough Performances

-  With the right conditions, “breakthroughs” in productivity are possible.
-  Solution is unknown when objective is established—e.g., JFK’s statement that the U.S. would put a man on the moon within a decade.
-  Breakthroughs occur randomly (and rarely) in large shops, but are more common in small software organizations.
-  Possible computer-industry examples: Microsoft’s Windows-NT project, DEC’s Alpha project, etc.
-  Key question: What is required to **DELIBERATELY** instigate breakthrough performances in a large shop?




Identifying an extraordinary effort

-  **Must clearly define breakthrough project: one that meets a deadline or achieves a result previously thought unlikely, impractical, or even impossible.**
-  **Exceeds what's predictable or expected based on existing criteria for measurement.**
-  **Identifying characteristic: commitment to achieving a breakthrough without knowing how it will be done.**




Identifying an extraordinary effort

-  **Key factor in defining a breakthrough is time and effort invested by both senior and project-level management to assemble the team before start of project to emphasize its importance .**




Identifying an extraordinary effort

-  **Management must communicate to the team, in a credible fashion, why the breakthrough is needed.**
-  **Team members must understand why a breakthrough is so important—e.g., to avoid bankruptcy**
-  **...or even how achieving a successful result can enhance the reputation of the IT department within the organization.**







Identifying an extraordinary effort

-  Another motivating technique: frank discussion of negative consequences of not achieving superior productivity results on this particular project.
-  For example, perhaps department credibility is at stake, and users have threatened to outsource their work if IT cannot meet an extraordinarily tight deadline for producing the desired system.
-  For a startup company, it might mean not getting the next round of funding — or watching Microsoft bring out their version of the product before you can release yours.







Identifying an extraordinary effort

-  This kind of communication rarely occurs — especially in big companies.
-  Often there is no formal kickoff meeting.
-  Managers may not have mastered necessary communication techniques to empower the team with the required sense of mission and to overcome skepticism that this is merely “the company line.”





More suggestions

-  **Give team members a choice of whether to participate.**
 -  Just like the [Mission Impossible](#) instructions: “Your mission, if you choose to accept it...”
 -  Irrelevant if decision not to participate leads to demotion, firing, insults, etc.
-  **Look for ways for the team to make a commitment to a vision**
-  **Watch for breakdowns— frequently lead to breakthroughs.**
-  **Recognize and accept the greater degree of risk in this approach.**





5.3 The importance of communications with the team

-  **Ideal strategy: the project manager has no secrets**
-  **Crucial that everyone knows current information on priorities, politics, constraints, risks, etc.**
-  **Communication between team members is also crucial, especially if members have not worked together before**
-  **Intra-team communication has to be kept confidential (from outsiders) to encourage honesty and frankness**
-  **Strongly implies the need for e-mail and probably more sophisticated features — e.g., Lotus Notes to keep ongoing “threads” of discussions, etc.**
-  **Weekly lunch/beer/dinner sessions likely to be crucial, too**

5.4 Team-Building Issues

-  **Question: how would you recognize an effective e-business project team**
-  **Be aware of team roles — e.g., importance of visionary, skeptic, team worker, etc.**
-  **Consider using Briggs-Meyers profiles so people will know how they'll react to each other. Especially relevant in e-business projects, because there is often a mix of IT- and non-IT personalities.**
-  **Watch out for “teamicide” issues often caused by project stress**

Signs of an Effective Team

-  ***Strong sense of identity.*** Team names, common “in-jokes,” common lunch area, etc.
-  ***Sense of eliteness.*** Team members feel they are part of something unique.
-  ***Joint ownership:*** Participants are pleased to have their names grouped together on a product.
-  ***Obvious enjoyment.*** Do good work and have fun.

One issue: team personalities

The Briggs-Meyers personality type model

		SENSING TYPES S		INTUITIVE TYPES N	
		THINKING T	FEELING F	FEELING F	THINKING T
I N T R O V E R T S	J U D G I N G J	ISTJ Serious, quiet, earn success by concentration and thoroughness. Practical, orderly, matter-of-fact, logical, realistic, and dependable. Take responsibility.	ISFJ Quiet, friendly, responsible, and conscientious. Work devotedly to meet their obligations. Thorough, painstaking, accurate. Loyal, considerate.	INFJ Succeed by perseverance, originality, and desire to do whatever is needed, wanted. Quietly forceful; concerned for others. Respected for their firm principles.	INTJ Usually have original minds and great drive for their own ideas and purposes. Skeptical, critical, independent, determined, often stubborn.
	P E R C E I V I N G P	ISTP Cool onlookers—quiet, reserved, and analytical. Usually interested in impersonal principles, how and why mechanical things work. Flashes of original humor.	ISFP Retiring, quietly friendly, sensitive, kind, modest about their abilities. Shun disagreements. Often relaxed about getting things done.	INFP Care about learning, ideas, language, and independent projects of their own. Tend to undertake too much, then somehow get it done. Friendly but often too absorbed.	INTP Quiet, reserved, impersonal. Enjoy theoretical or scientific subjects. Usually interested mainly in ideas, little liking for parties or small talk. Sharply defined interests.
E X T R O V E R T S	P E R C E I V I N G P	ESTP Matter-of-fact, do not worry or hurry, enjoy whatever comes along. May be a bit blunt or insensitive. Best with real things that can be taken apart or put together.	ESFP Outgoing, easygoing, accepting, friendly; make things more fun for others by their enjoyment. Like sports, making things. Find remembering facts easier than mastering theories.	ENFP Warmly enthusiastic, high-spirited, ingenious, imaginative. Able to do almost anything that interests them. Quick with a solution and to help with a problem.	ENTP Quick, ingenious, good at many things. May argue either side of a question for fun. Resourceful in solving challenging problems but may neglect routine assignments.
	J U D G I N G J	ESTJ Practical, realistic, matter-of-fact, with a natural head for business or mechanics. Not interested in subjects they see no use for. Like to organize and run activities.	ESFJ Warm-hearted, talkative, popular, conscientious, born cooperators. Need harmony. Work best with encouragement. Little interest in abstract thinking or technical subjects.	ENFJ Responsive and responsible. Generally feel real concern for what others think or want. Sociable, popular. Sensitive to praise and criticism.	ENTJ Hearty, frank, decisive, leaders. Usually good in anything that requires reasoning and intelligent talk. May sometimes be more positive than their experience in an area warrants

A Common Personality Type

ISTJ Serious, quiet, earn success by concentration and thoroughness. Practical, orderly, matter-of-fact, logical, realistic, and dependable. Takes responsibility.









Another Common Personality Type

ESTJ Practical, realistic, matter-of-fact, with a natural head for business or mechanics. Not interested in subjects they see no use for. Likes to organize and run activities.




The Ideal Manager...

ENFJ Responsive and responsible. Generally feel real concern for what others think or want. Sociable, popular. Sensitive to praise and criticism.

The team role model: 8 critical roles

-  **Chairman**
-  **Shaper**
-  **Plant**
-  **Monitor-Evaluator**
-  **Company Worker**
-  **Team Worker**
-  **Resource Investigator**
-  **Completer**

Typical distribution of personalities

PERSONALITY TYPE	POPULATION	TEAM ROLES 	
ISTJ	247 (38%)	CW,ME,CE,TS	Note relative absence of feeling (F) and perceiving (P) people, and predominance of introverts.
ESTJ	163 (25%)	CW,ME,CF,TS	
ENTJ,INTJ	42, 43		 Project members typically describe ideal project manager with an ENFJ personality— <i>opposite</i> of their personalities.
ISFJ, ESFJ,ENFJ	34,32,35		
INFJ	19		
ESFP, ENFP, ENTP	10,12,11		 Consequence: computing teams lack interpersonal skills , as well as assertiveness and negotiation skills.
ISTP, ESTP,ISFP,	5,4,2		
INTP,INFP	4,3		

CW = company worker
 CF = completer
 ME = monitor-evaluator
 TS = technical shaper








From Rob Thomsett's study of approximately 650 Australian software engineers

CALLING IT A TEAM DOESN'T MAKE IT SO







“You can’t make teams jell. You can hope they will jell; you can cross your fingers; you can act to improve the odds of jelling; but you can’t make it happen. The process is much too fragile to be controlled.”

Tom DeMarco and Tim Lister, *Peopleware*

TEAMICIDE

-  **Defensive management—not trusting the team**
-  **Bureaucracy—too much paperwork**
-  **Physical separation of team members**
-  **Fragmentation of people's time**
-  **Quality reduction of the product**
-  **Phony deadlines**
-  **Clique control—splitting up teams**






Issues in forming effective teams

-  **Context:** How can “the way we do things around here” affect team performance?
-  **Management:** How should teams be directed? Who leads? Who is responsible? How are conflicts resolved?
-  **Vision:** How does the group come to understand and articulate the application problem? How does the group invent, disseminate, and validate the solution?
-  **Structure:** How should roles, responsibility, and decision making be organized?
-  **Personality:** How can basic working styles and roles impact team performance?
-  **Technology:** How can methods and tools affect team performance?










Different Team Structures

Broad objective	Degree of teamwork	Dominant theme	Process Focus	Member selection criteria	Examples
Problem solving	Situational	Trust, working relationship <i>get the process right</i>	Issues	Intelligence, sensitivity	CASE tool selection project
Creative	Spontaneous	Autonomy from rules, bureaucracy <i>get the people right</i>	Possibilities and alternatives	Independent thinkers, self-starters	Application development team
Tactical	High, systematic	Clear and comprehensive planning, <i>get the plan right</i>	Directions, well-defined operational standards, role clarity	Loyal, responsive, conforming	Software maintenance team





The Evolution of a team

-  **Forming:** team members define goals, roles, and direction of the team
-  **Storming:** team sets rules and decision-making processes, often renegotiates (argues) over team roles and responsibilities.
-  **Norming:** Procedures, standards, and criteria are agreed upon
-  **Performing:** the team begins to function as a system
-  this applies also to the vision-building process of developing a shared understanding of the application problem and the general structure of the solution

Team-building recommendations

-  Match the team structure to the technical problem and available people
-  Match degree of self-direction to norms of the organization, the structure of the team, and the abilities of the players. A self-managed team may not be practical
-  Definition & assignment of roles are essential, as is the expectation that the team make and meet commitments for time, cost, and quality.
-  Allow for "ramp-up" time. Teams do not "gel" without healthy conflict, and effective teams evolve their own ways for resolving conflict.
-  Match the players to the tasks.
-  Pay attention to the needs of each individual. Provide performance feedback, recognition, and reward.
-  Develop, communicate, and validate a vision of the application problem, its solution, and the system architecture.
-  Select a modeling language and appropriate automated support to facilitate communication of the team vision.
-  Recognize that teams are dependent on their environment. Develop a systems dynamic model to better understand how teams work.

4.5 Manager's Role in a Project Team

-  **Strategy #1: manager is a player-coach, living and working “in the trenches” with the troops**
-  **Strategy #2: manager is a “buffer” who keeps outside forces away, and leaves the team alone to get their work done**
-  **Strategy #3: manager is an evil ogre who bullies and terrifies the team into superhuman efforts**
-  **Other strategies?**




An Observation

“People under time pressure don’t work better; they just work faster.

In order to work faster, they may have to sacrifice the quality of the product and their own job satisfaction.”

Tom DeMarco and Tim Lister, [Peopleware](#)

The impact of workplace on quality





-  **Workers who say their workplace is acceptably quiet are 1/3 more likely to deliver zero-defect work.**
-  **66% of zero-defect workers report noise-level is OK in their workplace**
-  **8% of one-or-more-defect workers report noise level is OK in their workplace**

Best and Worst Performers in the Coding War Games




Environmental Factor	1st Quartile	4th Quartile
1. How much dedicated workspace do you have?	78 sq ft	46 sq ft
2. Is it acceptably quiet?	57% yes	29% yes
3. Is it acceptably private?	62% yes	19% yes
4. Can you silence your phone?	52% yes	10% yes
5. Can you divert your calls?	76% yes	19% yes
6. Do people often interrupt you needlessly?	38% yes	76% yes

Tom DeMarco and Tim Lister, [Peopleware](#)





More on office space

-  **Ideal: private 100 sq-ft office with 30 square feet of work surface for personal workstation, white board, file cabinets, workspace, plus workstation at home.**
-  **Example: Microsoft, where all programmers have a private office, with a door that closes, and a permanent phone number that follows them if they get moved to a different office**
-  **Tolerable: cubicles with 6-ft partitions & soundproofing.**
-  **Intolerable but common: waist-high partitions, stale Muzak, phones ringing, people yelling, dogs barking.**


More on office space

-  **“Police-mentality planners design workplaces the way they would design prisons: optimized for containment at minimal cost. We have unthinkingly yielded to them on the subject of workplace design, yet for most organizations with productivity problems, there is no more fruitful area for improvement than the workplace.**
-  **As long as workers are crowded into noisy, sterile, disruptive space, it’s not worth improving anything but the workplace.”**
-  **Tom DeMarco and Tim Lister, [Peopleware](#)**

Office Space and Interruptions

-  Some software development work requires intense, uninterrupted periods of concentration.
-  Phone calls, etc require a period of “re-immersion” of 5-15 minutes. For a discussion of this, see Tom DeMarco’s new book, [Slack: Getting past burnout, busywork, and the myth of total efficiency](#)
-  Consider measuring “Environmental Factor”:
-  Uninterrupted-Hours / Body-Present Hours

Offices and Telephones

 **“It has come to my attention that many of you, when you are busy, are letting your phones ring for three rings and thus get switched over to one of the secretaries. With all these interruptions, the secretaries can never get any productive work done. The official policy here is that when you’re at your desk you will answer your phone before the third ring...”**

 **From an IT manager’s memo to his staff**





5.6. Establishing a proper peopleware “culture”

- ✍ **Key contributors to an overall peopleware “culture”:**
 - ✍ Recruiting
 - ✍ Selection of staff
 - ✍ Performance appraisal
 - ✍ Training
 - ✍ Compensation
 - ✍ Career development
 - ✍ Team and organization development
- ✍ **In the best case, this is part of the overall organizational culture — consistent from one project to another**
- ✍ **The SEI has developed a “peopleware maturity model” (CMM-PMM) to describe levels of maturity. See notes and discussions at [SEI P-CMM web site](#).**











The SEI's five levels of maturity in managing software human resources

- | | |
|----------------------------|---|
| 1 Herded | software personnel treated as a purchasable commodity |
| 2 Managed | management installs basic human resource practices |
| 3 Tailored | human resource practices tailored to software jobs |
| 4 Institutionalized | ...tied to organizational objectives |
| 5 Optimized | effectiveness of human resource practices compared to software project data and outcomes |





5.7 Project training and management “flight simulators”

-  Training for most project managers consists of two words: “good luck!”
-  Textbooks and/or seminars like this one are the next best thing
-  But consider something like the annual visit to the “flight simulator” required of commercial airline pilots
-  One good commercial example: “Project Challenge” from SHL Systemhouse, available on CD-ROM (alas, no longer available)

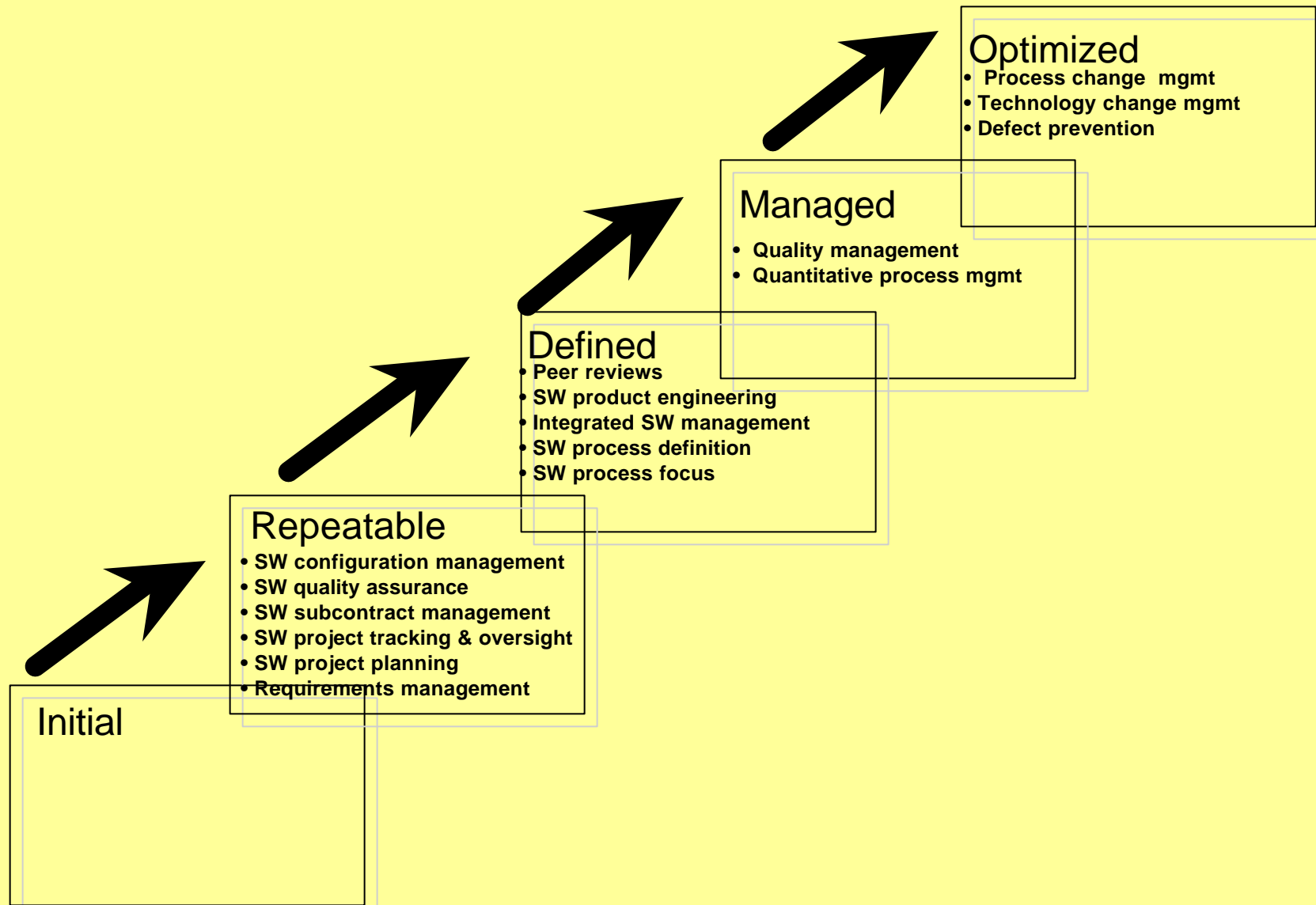
5.8 The role of “end user” manager in leading e-business projects

-  **Becoming more common in many companies**
-  **Traditional IT project manager may play a subordinate role, or end-user may direct all technical activities**
-  **Advantages**
 -  end-users knows their business needs better than anyone else
 -  end-users becoming more and more technically literate
 -  For high-press, mission-critical e-business projects: ensures that end-user department is really involved (like the pig), rather than standing on the side-lines and complaining about how lazy the programmers are
-  **Disadvantages**
 -  end-users are often busy running their business, don't have time to really participate — and this can be deadly in an e-business project.
 -  they often don't know how to manage technical people
 -  their technical knowledge is often superficial and dangerous

6. SOFTWARE PROCESSES

-  **Formal vs. informal processes**
-  **Importance of getting team to “own” their process**
-  **Importance of prototyping, RAD, and spiral methodologies**
-  **“Best practice” and “worst practice” concepts**

6.1 SEI vs. "lite" processes






6.1 “Lite” vs. “Heavy” Processes





- ✍ Formal (heavy) processes are great if you know what you’re doing...
- ✍ ...and if you’ve done the same thing several times before
- ✍ SEI-CMM guru Watts Humphrey: “if a process can’t be used in a crisis, it shouldn’t be used at all.”
- ✍ But many high-pressure projects involve doing things that have never been done before — with teams that have never worked together before.
- ✍ Conversely, if a team has worked together before, and really “jells”, then it doesn’t need a formal, heavy process
- ✍ Nevertheless, team needs to agree on what processes will be formalized (e.g., change management, source code control, testing(a la XP)), and what processes will be done on a completely ad hoc basis.
- ✍ For more details, see
 - ✍ “Extreme Programming,” by Jim Highsmith, [e-Business Application Delivery](#), Feb 2000.
 - ✍ November 2000 issue of [Cutter IT Journal](#) on “Light Methodologies”
 - ✍ “Put Your Process on a Diet,” by Martin Fowler, [Software Development](#), Dec 2000
 - ✍ “Retiring Lifecycle Dinosaurs,” by Jim Highsmith, [Software Testing & Quality Engineering](#), Jul/Aug 2000
 - ✍ “[The Light Touch](#),” by Ed Yourdon, Computerworld, Sep 18, 2000

6.1 More on “lite” versus “heavy”





Areas where there are differences

-  Degree of documentation
-  Frequency of reviews and approvals
-  Degree of decision-making authority — borrowed from “lean manufacturing” approach






Examples of documentation differences: the requirements analysis phase

-  Lite approach: one sentence per requirement
-  Medium approach: one paragraph per requirement
-  Heavy approach: detailed UML models, data dictionary, etc.
-  What happens to requirements when development is done?









Criteria for choosing lite vs heavy:

-  Project cost
-  Project duration
-  Staff size
-  Risk assessment — consequences of failure (safety-critical?)





6.1, cont'd SEI vs. "lite" processes

-  **"Heavy," SEI-style software processes are desirable(!) for nuclear reactors, pacemakers, air traffic control systems...**
-  **But unnecessary, impractical, and even undesirable in many high-pressure e-business projects...**
-  **What users really want is a system that's cheap enough, fast enough, feature-rich enough, and available soon enough — i.e., "good enough"**
-  **A project team oriented towards "good enough" software has some very different behaviors and strategies than conventional project teams**
-  **Software managers trained in traditional software engineering approaches have a hard time with this, but Silicon Valley lives with it every day.. As does anyone who interacts with the Web, where "dead" links and unloadable pages are part of the "Internet experience."**





6.1, cont'd Why do we fail to achieve “good enough” software?

-  We have a tendency to define quality only in terms of defects
-  We assume that fewer defects = better quality, and we assume that “mo’ better” quality is always preferred by user.
-  We tend to define quality (defect) requirements/objectives once, at the beginning of the project and keep it fixed
-  We’ve been told for such a long time that processes are crucial, that we often forget that processes are “neutral” — a fool with a “process-tool” is still a fool
-  We pursue quality with a fixed process that we define once, at the beginning of the project (or, even worse, for all projects in the whole company)
-  We underestimate the non-linear tradeoffs between such key parameters as staff size, schedule, budget, and defects
-  We ignore the dynamics of the processes: time-delays, feedback loops, etc.
-  We ignore the “soft factors” associated with the process like morale, adequacy of office space, etc.








6.2 Prototyping, etc.

-  **Lots of variations: iterative, spiral, “scrum”, etc.**
-  **Usually advocated because user doesn’t know what his requirements really are.**
-  **But key reason for doing it in a high-pressure project is ability to demonstrate tangible results several times before the “ultimate deadline”**
-  **If prototypes have been chosen carefully, then version N-1 or N-2 can be delivered and put into production, if necessary**







6.3 An example of a “lite” process: OO-RAD

-  **What does OO-RAD mean?**
-  **Arguments in favor of rapid prototyping**
-  **Types of applications suited for OO-RAD**
-  **A recommended life cycle**

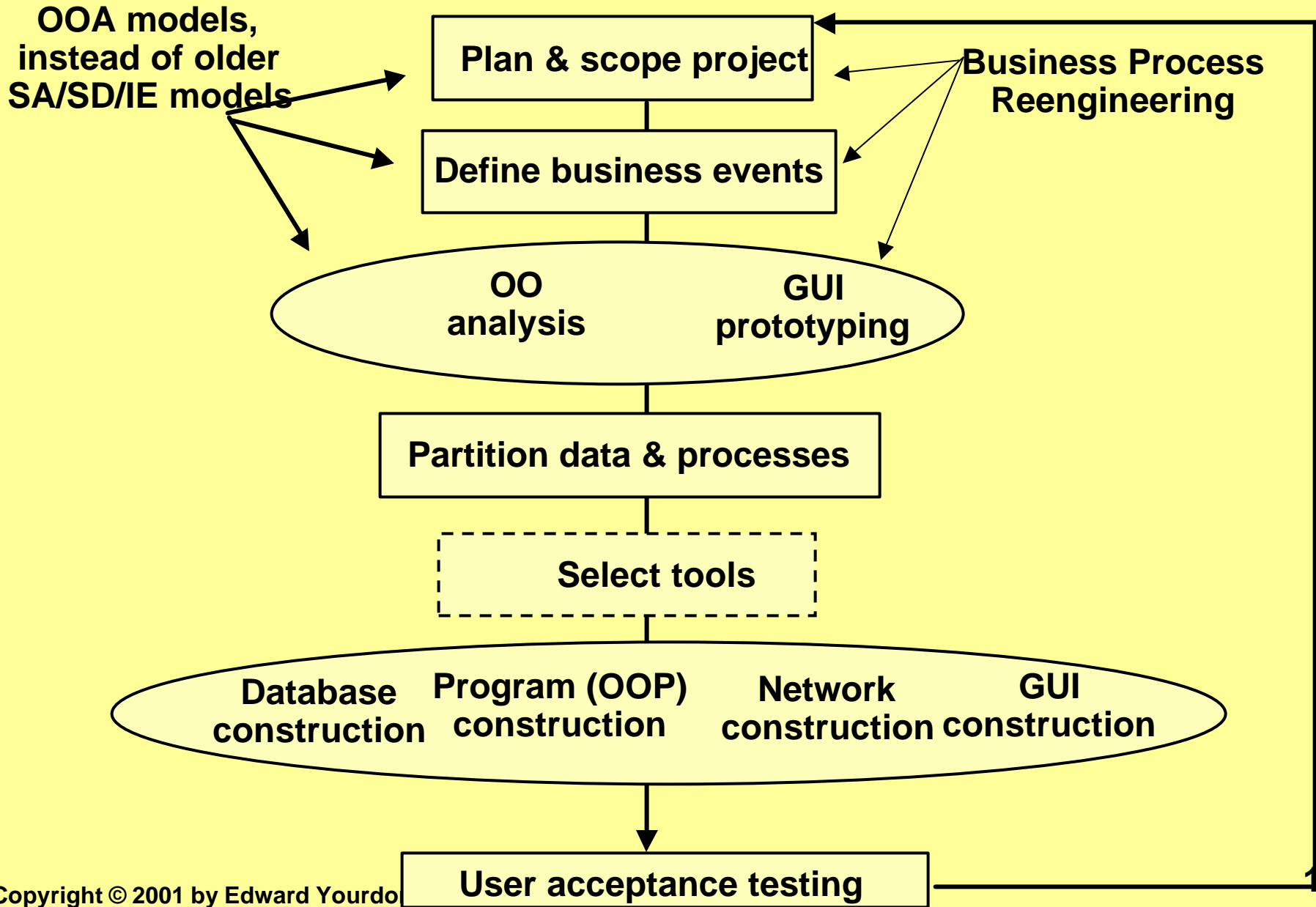
What does OO-RAD mean?

-  RAD = faster, implies prototyping, concurrent engineering approach, etc.
-  OO = “we programmed the whole thing in Java”
-  or OO = “we took advantage of a library of reusable components”
-  or OO = “we did a bunch of GUI stuff, ‘cause the users love windows and scroll bars”
-  or OO = client-server (?)
-  OO = some aspect of the OO paradigm of encapsulation, inheritance, communication via messages, polymorphism, etc.
-  OO-RAD = OO analysis + OO design + OO programming + RAD (but chances are that the user will only see the RAD part)

Types of application-development projects suited for OO-RAD

-  An important question: why does RAD have to be combined with OO?
-  (In case you don't believe that modeling is appropriate for e-business projects, see "The Importance of Business Modeling in e-Business," by Haim Kilov and Michael Guttman, [Cutter Consortium Distributed Computing Architecture/e-Business Advisory Service, March 2000, No. 2](#))
-  Projects where UI is the major factor (often true for any Web-based system, especially e-business) — which demands rapid prototyping, and usually involves OO-based implementation technology
-  Projects where productivity gain and "cycle time" gain depends on reuse — and specifically the need to create new subclasses of existing components
-  Projects where the nature of the application fits the "paradigm" of independent, encapsulated objects
-  Projects where someone believes that the "use case" OO paradigm will be the best way to model the user requirements.





Suggested OO-RAD Life Cycle



Products of OO-RAD life cycle

- ✍ A consolidated “product specification” often replaces old concept of separate analysis & design specifications
- ✍ More emphasis today on a separate requirements document, which may be a word-processing document that serves as input to the OOA model — and which may also be input to a requirements management tool, e.g.,
 - ✍ Caliber-RM: [Technology Builders](#)
 - ✍ DOORS: [Quality Systems & Software](#)
 - ✍ Requisite Pro: [Rational Software](#)
 - ✍ RTM Workshop: [Integrated Chipware](#)
 - ✍ Vital Link: [Compliance Automation](#)
- ✍ Typical products in a “formal” OO-RAD project (note that these are prototyped and evolved, too):
 - ✍ project plan
 - ✍ OOA object model, plus object specs
 - ✍ OOD object model, plus object specs
 - ✍ prototype objects - tuned and tested objects
 - ✍ Demo and Test Plan, plus test plan results

6.4 Getting the team to “own” the process

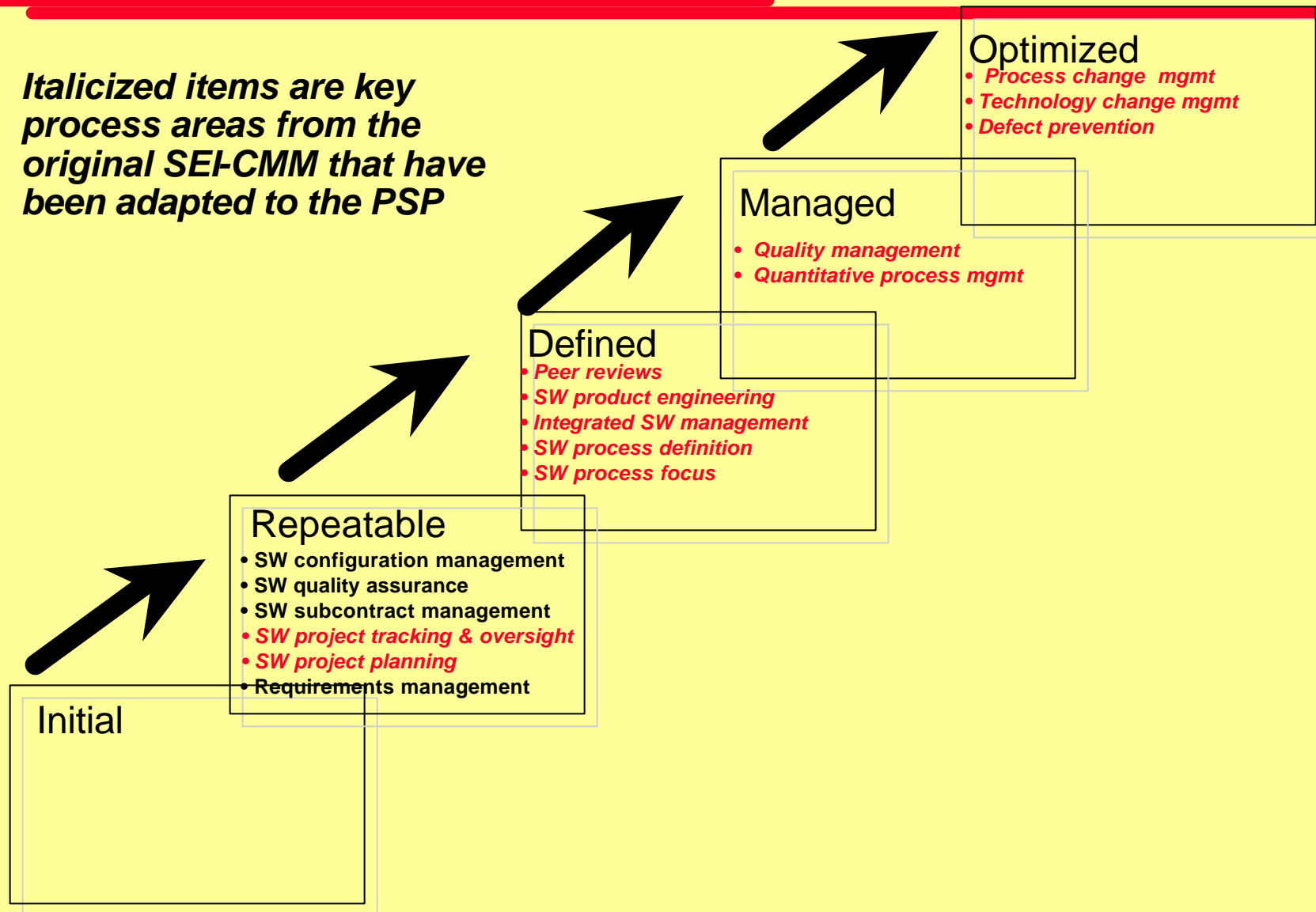
-  There's no point mandating a particular software process if it's not going to be followed
-  Which either means that project manager must impose it in a dictatorial fashion...
-  ...or the team must sincerely agree to adopt it, because they believe in it.
-  A corollary: it's usually a disaster to introduce a new, unfamiliar process in a project.

The Personal Software Process

- ✍ “Many viewed the CMM as designed for large organizations and did not see how it could be applied to individual work or to small project teams... The SEI thus started a process research project to examine ways individual software engineers could apply level 5 process principles. After several years of research, means were devised to adapt 12 of the 18 CMM key process areas to the work of individual software engineers.”
- ✍ Watts Humphrey: “A Personal Commitment to Software Quality,” American Programmer (now [*Cutter IT Journal*](#)) December 1994. See also [*An Introduction to the Personal Software Process*](#), by Watts Humphrey (Addison-Wesley, 1997)

The five levels of PSP maturity

Italicized items are key process areas from the original SEI-CMM that have been adapted to the PSP












PSP Experiment Results

Measurement area	Group A	Group B	Group C	Group D
Productivity Improvement	82.4%	22.9%	136.0%	61.6%
Defect Reduction	53.4%	45.8%	55.1%	80.1%

“...the problem is compounded by the fact that software engineers are rarely able to experiment. Everything they do is for delivery on a short and demanding schedule. An experiment would thus entail considerable risk. Not surprisingly, their reaction is to defer experimenting with new methods until they have some free time. Unfortunately, they never seem to have free time.”

6.5 The Airlie Council











“Principal Best Practices”

-  **Formal Risk management**
-  **Agreement on Interfaces**
-  **Peer Reviews**
-  **Metric-Based Scheduling and management**
-  **Binary Quality Gates at the “Inch-Pebble” Level**
-  **Program-Wide Visibility of Project Plan and Progress Vs. Plan**
-  **Defect Tracking Against Quality Targets**
-  **Configuration management**
-  **People-aware management Accountability**








6.5, cont'd Worst Practices

- ✍ Don't expect schedule compression of =10% compared to statistical norm for similar projects
- ✍ Don't justify new technology by the need for schedule compression
- ✍ Don't force customer-specific implementation solutions on the project
- ✍ Don't advocate the use of silver bullet approaches
- ✍ Don't miss an opportunity to move items that are under external control off the critical path
- ✍ Don't bury all project complexity in software as opposed to hardware
- ✍ Don't conduct critical system engineering tasks without sufficient software engineering expertise
- ✍ Don't expect to achieve an accurate view of project health from a formal review attended by a large number of unprepared, active reviewers
- ✍ Don't expect to recover from a schedule slip of =10% without acknowledging a disproportionately greater reduction in software functionality to be delivered.
- ✍ For more discussion along the same lines, involving the concept of “anti-processes,” see [*Anti-Patterns and Patterns in Software Configuration Management*](#), by William J. Brown, Hays W., Iii McCormick, Scott W. Thomas (Wiley, 1999).




6.5, cont'd Breathalyzer Test

-  Do you have a current, credible activity network supported by a work breakdown structure (WBS)?
-  Do you have a current, credible schedule and budget?
-  Do you know what software you are responsible for delivering?
-  Can you list the top ten project risks?
-  Do you know your schedule compression percentage?
-  What is the estimated size of your software deliverable? How was it derived?
-  Do you know the percentage of external interfaces that are not under your control?
-  Does your staff have sufficient expertise in the project domain?
-  Have you identified adequate staff to allocate to the scheduled tasks at the scheduled time?
-  All of this comes from the Airlie Council; for another discussion about the same concept, see “[Blowing the Whistle on Troubled Software Projects](#),” by Mark Keil and Daniel Roby, *Communications of the ACM*, April 2001







Other signs the project is in trouble

-  **Key team members quit**
-  **The Inverse Dilbert Correlation Factor**
-  **Excessive gallows humor**
-  **New names for the project — e.g., “Project Titanic”**
-  **An ominous silence from end-users and senior management, who used to ask on a daily basis how the project was coming along**
-  **Thrashing — lots of activity but no sign of forward progress**
-  **Visible presence of lots of “bean counters” and “suits” who were never there during the early, euphoric days**






7. MONITORING AND CONTROLLING PROGRESS

-  **Management styles for different types of software projects**
-  **Measuring, managing, and controlling progress**
-  **Maximizing project life cycle review**







7.1 Management styles for different types of projects

-  Even if you know precisely what the status of the project is, how do you communicate that to the team, and in what fashion do you use that information to change the team's behavior?
-  Theory X, theory Y, theory W, etc.
-  Note that a lot of this depends on choosing team members with appropriate personalities
-  Remember: you don't know the actual status of the project, and you don't know the actual delivery date — all you have is some degree of confidence in an estimate.
-  What kind of style is appropriate for a project that requires high innovation — but where something also has to be delivered eventually?
-  What about “traditional” development projects — or a maintenance project?

7.2 Measuring, managing, and controlling progress

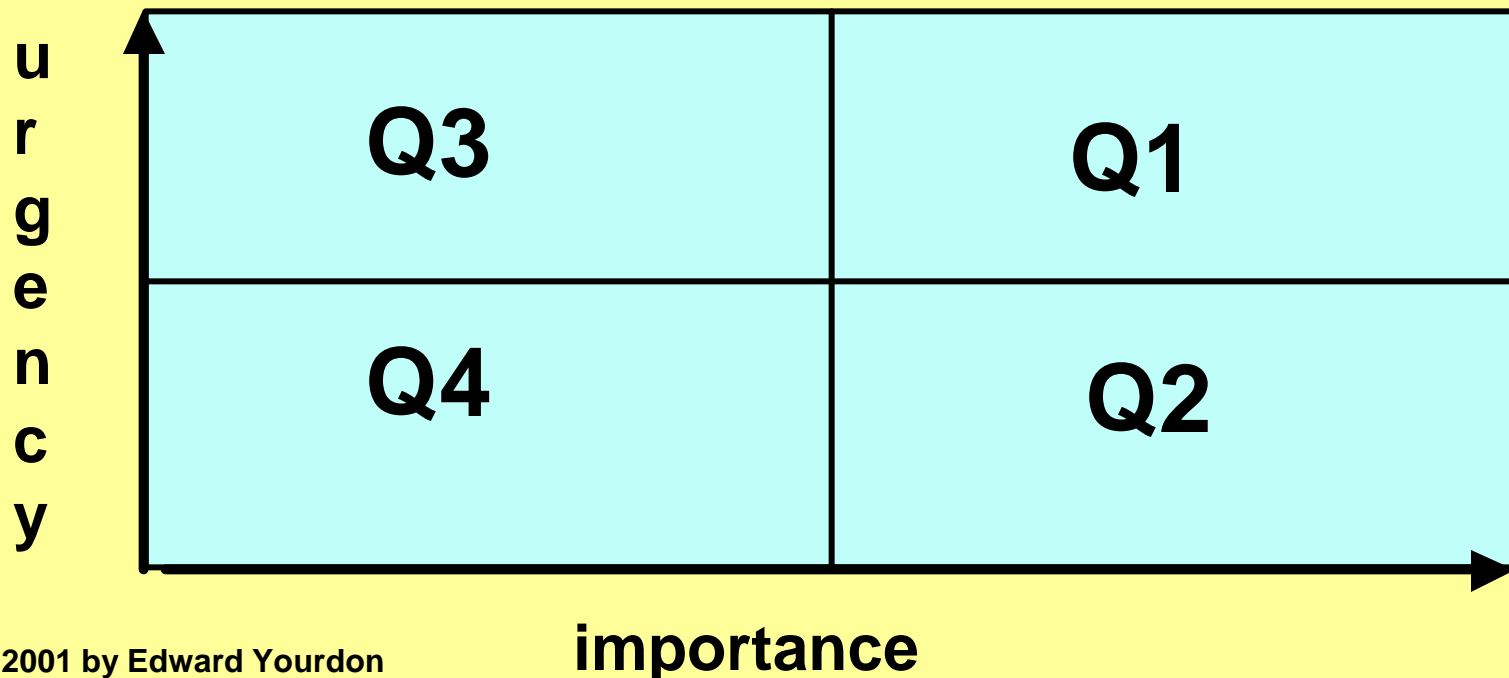
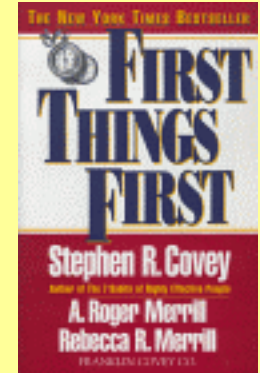
-  **General comments and suggestions**
-  **The importance of metrics and benchmarks**
-  **The importance of the “daily build” approach**
-  **Quality, testing, and defect management**
-  **Risk management**

7.2.1 General comments

-  Management approaches based on classical waterfall approach are almost certain to fail in large, complex projects
-  Need some kind of “time-box” approach based on versions, features, deliverables, etc.
-  Jim McCarthy: “Never let a programmer disappear into a dark room”
-  If team understands what features and dependencies are required for a milestone completion, they will exert their own pressure upon themselves, rather than having the manager beat them up.
-  If you miss one milestone deadline, it’s crucial to succeed on the next one.
-  Milestone post-mortems can be incredibly valuable.

7.2.1, cont'd General Comments










- ✍ Key objective for management: help your project team distinguish between urgent and important.
- ✍ Best reference: [*First Things First*](#), by Stephen R. Covey, et al (Fireside, 1996)
- ✍ Project manager should act as a buffer to keep Q3 items from distracting the project team.
- ✍ For more details, see my May 2001 *Computerworld* article, "[*Helping Developers Manage Their Time*](#)."







7.2.2 The “daily build”

- ✍ Popularized by Dave Cutler at Microsoft
- ✍ Jim McCarthy (former head of Microsoft’s Visual C++ project): “The daily build is the heartbeat of the project — it’s how you know you’re alive”
- ✍ Should be automated, and performed overnight — or even more often.
- ✍ Various “tricks” can be used to increase its effectiveness
 - ✍ Punishing people who “break” the daily build
 - ✍ Using red-flag/green-flag at office entrance








7.2.3 Maximizing project life cycle reviews

-  Formal reviews at lifecycle milestones are often used to gauge the progress and status of the project
-  But project manager must remember that these are often highly political events, in which the real problems and the real status is hidden in order to avoid political repercussions.
-  While you can't eliminate these formal reviews, consider augmenting them with informal, peer-level reviews
 -  See my [Structured Walkthroughs](#) (Prentice-Hall, 1989) for guidelines on procedures and suggestions for conducting such reviews
 -  ...or [Software Inspection](#), by Tom Gilb, Dorothy Graham, and Suzannah Finzi (Addison-Wesley, 1993) for discussion of formal inspections.
 -  ...or [Project Retrospectives: A Handbook for Team Reviews](#), by Norman Kerth (Dorset House, 2001) for an up-to-date perspective on reviews.
-  Most important: mini post-mortems with the team at the end of each inch-pebble deliverable.
-  See my *Computerworld* article, "[Mini-Postmortems](#)" for more detail.
-  Consider also the use of videotaping project walkthroughs and meetings in order to prevent future maintenance problems. For more details, see my article, "[Internet Legacy Nightmares](#)," *Computerworld*, Jul 31, 2000)










7.2.4 Binary Quality Gates at the “Inch-Pebble” Level

-  **Completion of each task in the lowest-level activity network needs to be defined by an objective binary indication.**
-  **Completion events should be “gates” that assess either the quality of the product produced, or the adequacy and completeness of the process that was carried out.**
-  **If planning is not in sufficient detail, any picture of how things are progressing is an illusion**
-  **We want to avoid the “90% done” syndrome that plagues many projects**







Binary Quality Gate “Best Practices” Essentials

-  **Ensure visibility of where the development really is based upon the products produced**
-  **Ensure that every lowest-level (inch-pebble) task:**
 -  **is of short duration**
 -  **expends only a small percent of the total budget**
 -  **is dedicated to producing a tangible product necessary for a required deliverable**
 -  **has a binary acceptance gate defined for it**
 -  **has no Earned Value credit assigned to it until the binary gate is passed**

Binary Quality Gate Status Checks








-  Are credible project status and planning estimates based on inch-pebble quality gates which can be aggregated at any desirable level produced?
-  Have all activities been decomposed into inch-pebbles?
-  Has all near-term work been decomposed into tasks no longer than two weeks in duration? (Maybe even one week!)
-  Have achievable accomplishment criteria been identified for each task? Are tasks based on overall quality goals and criteria for the project?
-  Are quality gates rigorously applied for determining task accomplishment, without exception?
-  Is there clear evidence that planned tasks are 100% complete before acceptance?
-  Is there clear evidence of successful completion of reviews?
-  Are inch-pebble tasks on the critical path defined, enabling more accurate assessment of schedule risks and contingency plans?
-  Is the set of binary quality gates compatible with the Work Breakdown Structure?

Defect Tracking Against Quality Targets












-  **The only way to keep project costs from exploding is to find and fix defects as they occur.**
-  **Cost of fixing defects typically increases tenfold as they pass into each subsequent development phase.**
-  **Thus, defects should be tracked formally at each project phase.**
-  **Configuration management should be used to record defects and trace them through to removal.**
-  **Remember that typically 20% of all software modules contain 80% of defects**
-  ***There is no such thing as a private defect* — i.e., one that is detected and removed without being recorded.**

Defect Tracking






“Best Practice” Essentials

-  **Establish a goal for delivered defects per unit of size (this includes requirements problems!)**
-  **Implement practices to find defects when they occur**
-  **Track average and maximum time to close a defect after it's reported**
-  **Track defect removal efficiency of:**
 -  **All defects found through all techniques reported to and tracked by CM**
 -  **All defects reported from field for the first year after installation of system**
-  **Grade developers on defect removal efficiency — e.g., defects found and fixed during development divided by defects found and fixed during first year of installation**






Defect Tracking Status Checks

-  Are defect targets established for the project? Are the targets firm?
-  Are consequences defined if a product fails to meet the target?
-  Do project quality targets apply to all products?
-  Are there circumstances defined under which quality targets are subject to revision?
-  What techniques are used to estimate latent defect counts?
-  How are current projected levels of defect removal empirically confirmed as adequate to achieve planned quality targets?
-  Is test coverage sufficient to indicate that the latent defect level achieved by the end of testing will be lower than the established quality targets?
-  Are the inspection and test techniques employed during the program effective in meeting quality targets?
-  Do all discovered defects undergo CM, and are accurate counts achieved for defects discovered and defects removed?
-  Is there a closed-loop system linking defect actions from when defects are first detected to when they're resolved?
-  Is the defect information defined at a level of granularity that supports an objective assessment of resolution on a periodic basis?






7.3 Risk Management

-  **Introduction and definition**
-  **The context of risk management**
-  **Critical cultural aspects**
-  **Risk assessment vs. risk control**
-  **Likely causes of risk in software projects**

Introduction and Definitions

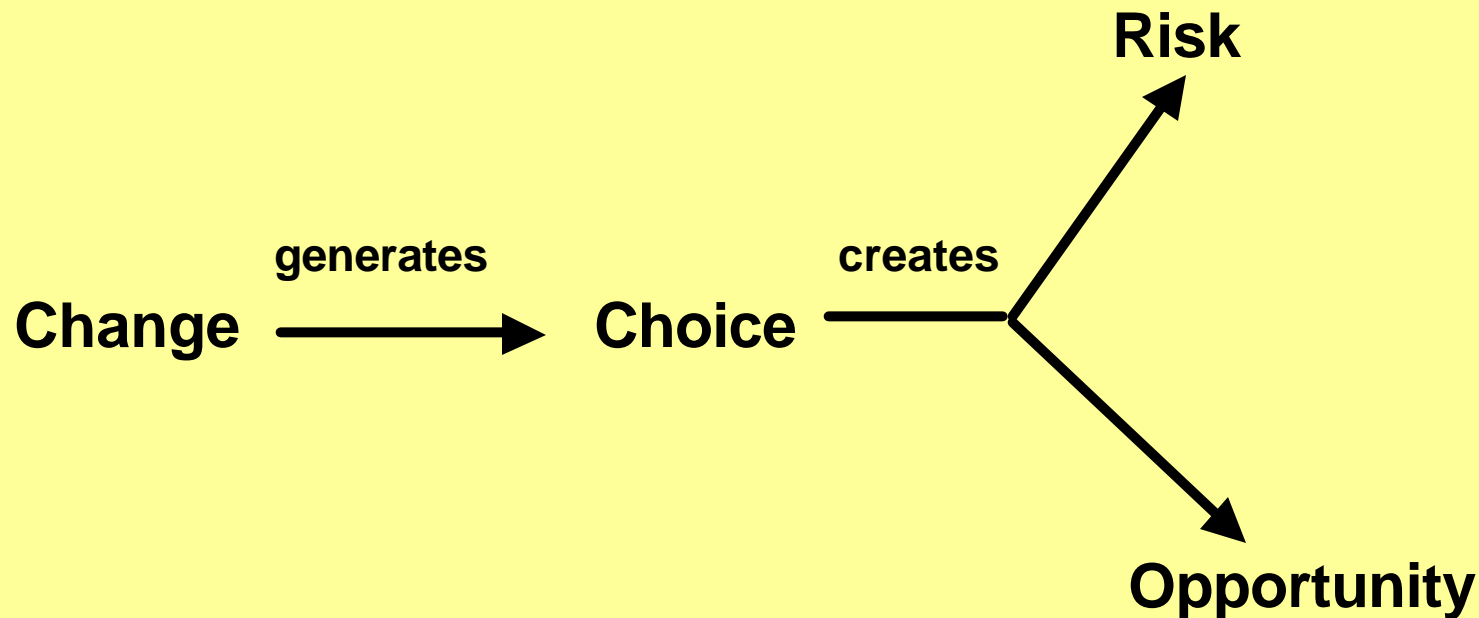
-  **Software risk management (SRM) is ignored or treated in an ad hoc fashion in most organizations today**
-  **Many software engineering experts believe that SRM will be one of the major “weapons” for mission-critical e-business projects**
-  **Quantitative metrics for the extent of common categories of software risks are beginning to be published**
-  **Hence, standardization and formalization of SRM is now becoming possible and realistic.**
-  **Often useful to distinguish between risk assessment, risk control, risk avoidance—all of which are components within the overall management process of SRM**

CONTEXT OF RISK MANAGEMENT

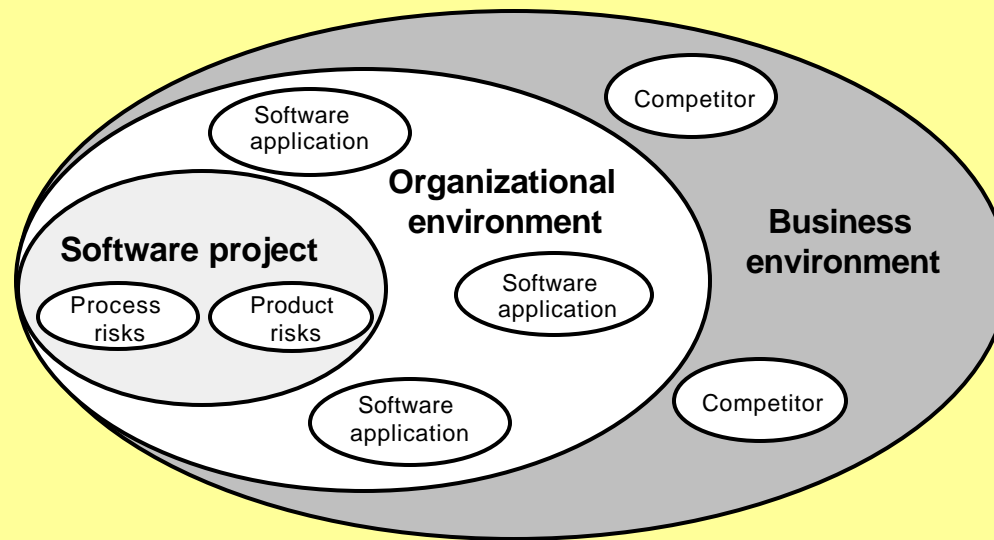
-  **SRM = managing change**
-  **SRM is part of a larger environment**
-  **SRM = organized aggression**
-  **What SRM tries to accomplish**
-  **Ways of managing risk**

SRM = Managing change

- ✍ If there is no **CHANGE** in the current situation, there may be no choices available. Unless a manager has a **CHOICE** in his decision-making process, then there is no **RISK**—for the future would be pre-determined.






SRM is part of a larger environment





- ✍ The major causes of project failures often exist in the organizational environment, or in the business environment—both of which are outside the scope of the software manager's jurisdiction.
- ✍ Conversely, sometimes project manager is unaware of the consequences of his software risks that are outside the software project per se.
- ✍ So SRM should be viewed as a dynamic, competitive process rather than a static defensive once-only procedure.

What SRM tries to accomplish

MINIMIZE UNCERTAINTY

-  Operational information—information required to make the right business decisions, e.g., the right estimates of time and budget for projects
-  Operational control—the degree of influence a group has over its own destiny, e.g., standards, organizational structures.
-  Operational time—how rapidly a group can respond to internal/external stimuli, e.g., timely marketing information.

MAXIMIZE STABILITY AND PREDICTABILITY

-  from both an internal...
-  and external perspective.

Ways of managing risk






 **Reactive**

 **Transitional**




 **Proactive**

Most projects are reactive — but you should be aware of your situation, and making plans to move to a more pro-active approach






Reactive Risk Management

-  The organization reacts to risks as they occur
-  Best form: mitigation of risk symptoms—alleviating a risk's consequences by providing additional resources (e.g., more testing)
-  Less desirable: “fix on failure” approach, where risks are addressed after they have surfaced
-  Least desirable: crisis mode, “fire fighting,” which typically indicates that the organization is fighting for survival.
-  Risk prevention is usually a transition point between a reactive and proactive approach—e.g., planning during the early stages of a project so that risks can be precluded from occurring.

Proactive Risk Management

-  **Elimination of root causes of failure—e.g., total quality management approaches.**
-  **Expanding the scope of risk assessment to a broader horizon, to allow anticipation of failure and risk**
-  **Management of change—developing a risk-taking ethic by engineering risk and opportunity.**





Critical Cultural Issues

-  **There must be a standard process for assessing and managing risks, consistent across all projects**
-  **Must be able to support different organizational perspectives — what engineering sees as a risk, marketing may see as an opportunity**
-  **SRM should support the development of a risk-taking ethic, to encourage proactive rather than reactive management**
-  **SRM must work hard to avoid the common phenomenon of “shooting the messenger” who reports a risk—especially for working-level people.**
-  **A corollary to all this: consultants usually know that they can get a better assessment of the real status of a project by talking to the people in the trenches**





Risk Assessment vs. Control

- ✍ Risk assessment is usually done by considering the following aspects of a project:
 - ✍ System/product complexity
 - ✍ Client or target environment
 - ✍ Team environment
- ✍ System complexity assessment may include function points, level of programming language, performance constraints, etc.
- ✍ Client complexity assessment may include number of users involved, level of user knowledge, priority of system within user area, need for reorganization etc
- ✍ Team environment assessment typically includes capability, morale, and experience of team members, as well as software process used by the team. See “Mitigating Business Risk Through Software Capability Improvement,” by Colin Tully, [Cutter Consortium Business-IT Advisory Service](#), Executive Report, Vol. 3, No. 12
- ✍ An interesting assessment from Jim Highsmith, “High speed is easy; high change is the real challenge.” See Chapter 7 of Highsmith’s [Adaptive Software Development: A Collaborative Approach to Managing Complex Systems](#)









Subjective vs. Objective Assessment

-  Typically, there are >100 risk factors that could be included in a risk model
-  Some may be easily quantified, e.g., software size, measured by function points
-  For other factors (e.g., degree of user hostility), a qualitative assessment, developed with a group consensus, is typically the only practical approach.
-  For practical management approach, usually advisable to categorize all individual risks (as well as aggregate sum of risks) as “high,” “medium,” and “low.”

Risk Control

-  **Project manager and team can often identify strategies to minimize or eliminate many risk factors.**
-  **High-risk factors that cannot be summarily eliminated should be documented with a “risk memorandum” that identifies the risk impact, possible higher-level actions, contingency plans, etc**
-  **Particularly important because (as noted earlier) many risks are beyond the control of project team to deal with**
-  **A simple but excellent approach in this area: monitoring the “top ten” list of risks in regular project management status review meetings.**

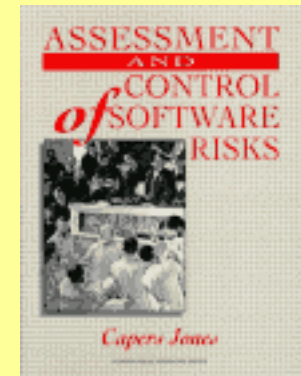
Risk Impact

-  **Financial**—investment is lost, benefits not gained
-  **Strategic**—company's strategic plan is compromised
-  **Technical**—technology platforms compromised
-  **Legal**—vulnerability to lawsuits, etc
-  **Political**—violation of government regulations
-  **Commercial**—loss of market share
-  **Fraud**—exposure to fraud and security violations
-  **Image**—loss of public reputation, etc.

Likely Causes of Software Risks

✍ Are now being identified for major categories of software projects:

- ✍ MIS projects
- ✍ Systems software projects
- ✍ Commercial software products
- ✍ Military software projects
- ✍ Outsourced/contract projects
- ✍ End-user software projects
- ✍ E-business projects



✍ See [Assessment and Control of Software Risks](#), by Capers Jones (Prentice Hall, 1994)

✍ See also [Managing Risk: Methods for Software Systems Development](#), by Elaine M. Hall (Addison-Wesley, 1998)

✍ See also “Risk Management for E-business,” by Carole Edrich, [Cutter Consortium Distributed Computing Architecture/E-Business Advisory Service](#), Executive Report, Vol. 3, No. 12

MIS Systems Risks

RISK FACTOR	% OF PROJECTS AT RISK
Creeping user requirements	80
Excessive schedule pressure	65
Low quality	60
Cost overruns	55
Inadequate configuration control	50

System Software Risks

RISK FACTOR	% OF PROJECTS AT RISK
Long schedules	70
Inadequate cost estimating	65
Excessive paperwork	60
Error-prone modules	50
Canceled projects	35

Commercial software risks

RISK FACTOR	% OF PROJECTS AT RISK
Inadequate user documentation	70
Low user satisfaction	55
Excessive time to market	50
Harmful competitive actions	45
Litigation expense	50

Military software risks

RISK FACTOR	% OF PROJECTS AT RISK
Excessive paperwork	90
Low productivity	85
Long schedules	75
Creeping user requirements	70
Unused or unusable software	45







Contract/outourced software risks

RISK FACTOR	% OF PROJECTS AT RISK
High maintenance costs	60
Contractor/client friction	50
Creeping user requirements	45
Unanticipated acceptance criteria	30
Legal ownership of software/deliverables	50




End-User Software Risks

RISK FACTOR	% OF PROJECTS AT RISK
Non-transferable applications	80
Hidden errors	65
Unmaintainable software	60
Redundant applications	50
Legal ownership of software/deliverables	50





E-business Risks

-  Summarized and discussed in “Risk Management for E-business,” by Carole Edrich, [Cutter Consortium Distributed Computing Architecture/E-Business Advisory Service](#), Executive Report, Vol. 3, No. 12
-  Most e-businesses are subject to immediate international access — which raises potential legal and regulatory risks
-  Internet supply chains behave very differently than those in a brick-and-mortar organization — risk of increased exposure to unstable suppliers.
-  Pressure to be “first to market” in Internet space often leads to hasty decisions, creating technology risks, as well as risks associated with sales and marketing
-  Customer expectations are different (and often higher), loyalty is typically much lower.
-  Intellectual capital is more difficult to protect, when it can be accessed on your web site by competitors.








8. LANGUAGES, TOOLS, AND TECHNOLOGY

-  **Identifying a minimal toolset**
-  **A checklist of tools**
-  **Risks of introducing new, unfamiliar tools**







8.1 Identifying a minimal toolset

-  A high-pressure project must be allowed to choose its own tools, regardless of whether it conforms to organizational standards
-  ...but team members need to agree on common tools within the project — otherwise, chaos will occur
-  Unless team has worked together before on several projects, this implies a “minimal” set of tools that everyone will use
-  For more on this topic, see my December 2000 *Computerworld* article, “[The Value of Tools](#)”









8.2 Tool checklist

-  **Collaboration tools**
-  **Prototyping/RAD development tools**
-  **Configuration management/version control**
-  **CASE tools for analysis/design**
-  **Testing, debugging tools**
-  **Project management (estimating, scheduling, PERT/GANTT, etc)**
-  **toolbag of reusable components**

8.2.1 Collaboration tools

-  **Note: email is part of the standard “infrastructure” in virtually every organization today**
-  **Strongly encouraged: Lotus Notes, VaxNotes, or any other “threaded” groupware facility to maintain archives of technical discussions during project.**
-  **Also, check out such groupware tools as [eproject](#), [egroups](#), [ezboard](#), etc. Also, take a look at a new tool called “zaplets,” available from [Zaplet, Inc.](#)**
-  **Some additional collaboration/project-support websites: [eroom.net](#), [done.com](#), [hotoffice.com](#).**
-  **See July 2000 issue of [Fast Company](#) for discussion of these three. Also, see “Collaborative Project Management,” by Jim Highsmith, [e-Business Application Delivery](#), April 2000**
-  **Make sure that team members have access to email at home, while traveling, etc.**

8.2.2 Prototyping/RAD tools

-  This is often influenced by the company standards for tools, languages, etc.
-  Basic point: almost every language has a “visual” RAD-oriented development environment available today.
-  Typical examples:
 -  Microsoft’s Visual Studio
 -  Borland’s Delphi and Java environments
 -  IBM’s Visual Age for Smalltalk, COBOL, Java
 -  Powersoft’s PowerBuilder
 -  Various Web-site/e-business tools, e.g., [Broadvision](#)’s templates










8.2.3 CM Tools

- ✍ **Crucial to maintain version control and management of all changes and updates**
- ✍ **Remember that this includes more than just source code — you want to provide integrated control of all project artifacts**
- ✍ **Typical examples (see Sep 97 issue of Cutter's [e-Business Development Strategies](#))**
 - ✍ **first-tier: INTERSOLV's PVCS and Pure Atria's ClearCase.**
 - ✍ **second-tier: Integrated Systems, Inc.'s MatrixX, and Microsoft' Visual SourceSafe.**
 - ✍ **third-tier: Platinum's CCC/HARVEST, MKS's WebIntegrity and SourceIntegrity, Starbase's Starteam, Select Software Tools' Visual Enabler**












8.2.4 CASE tools for analysis

- ✍ This assumes that you've decided to formalize analysis/design with structured methods or OO methods
- ✍ Beware the learning curve, and the possibility your team isn't familiar with methodology
- ✍ Unless team has prior experience, stay away from the high-end, sophisticated tools
- ✍ Good examples
 - ✍ [Popkin](#) System Architect
 - ✍ [Visible Systems](#) Visible Analyst
 - ✍ [Rational](#) ROSE
 - ✍ Platinum's Paradigm Plus
 - ✍ [Togethersoft's](#) Together/C++

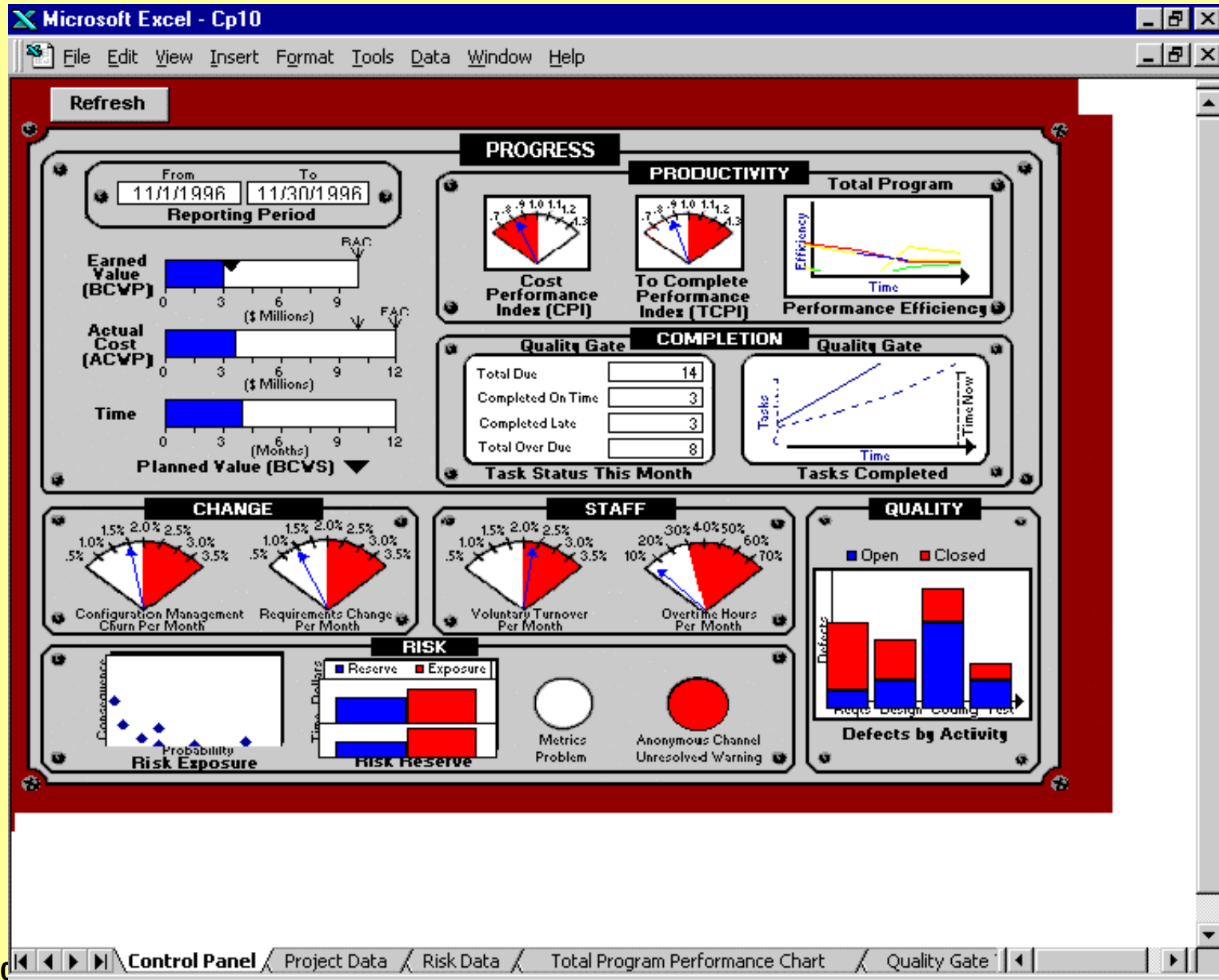
8.2.5 Testing Tools

-  **Fundamental point: you need to automate as much of testing as possible**
-  **Also want to integrate testing tools with configuration management, defect-tracking etc.**
-  **Particularly important for client-server, Internet systems; also for load-testing**
-  **Good examples**
 -  **SQA/Rational's ROBOT**
 -  **Mercury Interactive**
 -  **Platinum Software's Tools**
 -  **AutoTester**
 -  **Software Research**

8.2.6 Project Mgmt Tools

-  **Primarily used for scheduling, estimating at beginning of project...**
-  **...followed by project tracking and control throughout the project**
-  **Look for tools that provide a “dashboard” metaphor for reporting; see next page for an example**
-  **Popular/leading products in this area:**
 -  **Microsoft Project**
 -  **Digital Tools' AutoPLAN**
 -  **ABT's Project Management Workbench**
 -  **SAS Software's project management tools**
 -  **Primavera's SureTrack project manager**
 -  **CA SuperProject**
 -  **Platinum's Process Continuum**





8.2.6 Project Mgmt Tools, cont'd



8.2.7 Reusable components

- ✍ This is very definitely language/platform dependent
- ✍ Also involves significant architectural decisions — e.g., CORBA vs. COM/DCOM. (See “Components for the e-Business Age,” by Paul Harmon, [Component Development Strategies](#), March 2000)
- ✍ Often vendor-dependent, too — e.g., Microsoft’s MFC library vs. C++ libraries from other vendors (Rogue Wave, etc.)
- ✍ An interesting new area: industry-specific reusable “frameworks” (see IBM’s “[San Francisco](#)” project)
- ✍ Key point: you won’t get much of a productivity benefit from these libraries unless your project team is already familiar with them!

8.3 Risks of choosing new tools

-  **Some projects grab new tools as a “silver bullet” to accomplish much higher levels of productivity than would otherwise be possible...**
-  **...but they ignore the learning curve, confusion, and political debates associated with the introduction of new tool**
-  **And the tools are often so new that they don't even work properly yet**
-  **An irony: new tool sometimes is the straw that breaks the camel's back — and project failure is then blamed on the tool**

Words to live by in the software field

“I wake up each morning determined to change the World ...

...and also to have one hell of a good time.

Sometimes that makes planning the day a little difficult.”

E.B. White

found in the opening of the preface of

[Succeeding with Objects](#),

by Adele Goldberg and Kenneth S. Rubin

(Addison-Wesley, 1995)

REFERENCES

Tom DeMarco, [*The Deadline: A Novel About Project Management*](#) (Dorset House, 1997)

Tom DeMarco, [*Slack: Getting past burnout, busywork, and the myth of total efficiency*](#) (Broadway Books, 2001)

Jim McCarthy, [*Dynamics of Software Development: 'Don't Flip the Bozo Bit' and 53 More Rules for Delivering Great Software on Time*](#) (Microsoft Press, 1995)

Ed Yourdon, [*Death March*](#) (textbook edition, Prentice Hall, Jun 1999)

Larry L. Constantine, [*The Peopleware Papers: Notes on the Human Side of Computing*](#) (Prentice Hall/Yourdon Press, 2001)

John Boddie, *Crunch Mode* (Prentice Hall, 1987) (out of print)

Tom DeMarco and Tim Lister, [*Peopleware*](#) (2nd edition, Dorset House, 1997)

Tracy Kidder, [*The Soul of a New Machine*](#) (Back Bay Books, Jun 2000)

Stephen R. Covey, [*First Things First*](#) (Fireside, 1996)

Alan M. Davis, [*201 Principles of Software Development*](#) (McGraw-Hill, 1995)

Fred Brooks, [*The Mythical Man-Month*](#) (20th anniversary edition, Addison-Wesley, 1995)