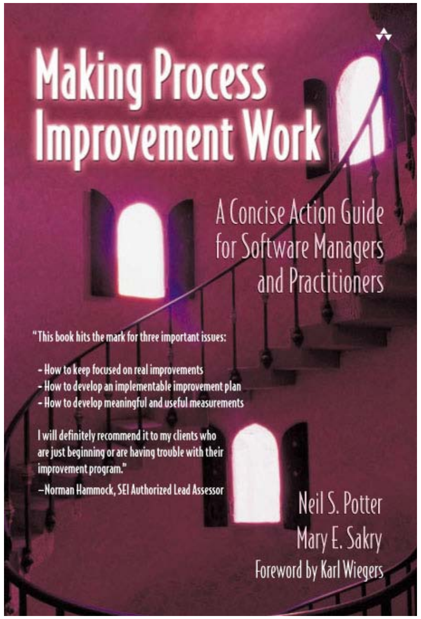
# Making Process Improvement Work

Tying Improvement and CMMI® Directly to What You Care About

**Neil Potter**

**Mary Sakry**

The Process Group
help@processgroup.com
www.processgroup.com



Making Process Improvement Work

A Concise Action Guide for Software Managers and Practitioners

"This book hits the mark for three important issues:

- How to keep focused on real improvements
- How to develop an implementable improvement plan
- How to develop meaningful and useful measurements

I will definitely recommend it to my clients who are just beginning or are having trouble with their improvement program."
–Norman Hammock, SEI Authorized Lead Assessor

Neil S. Potter
Mary E. Sakry
Foreword by Karl Wiegers

# Agenda - 1

1. **Introduction……………………………………………...**
2. **Developing a Plan………………………………….**
   - Scope the Improvement……………………………………...
   - Exercise……………………………………………….…
   - Develop an Action Plan……………………………………

# Agenda - 2

3. **Implementing the Plan…………………………………..**
   - Sell Solutions Based on Needs……………………………….
   - Work with the Willing and Needy First………………………..
4. **Checking Progress……………………………………..**
   - Are We Making Progress on the Goals? …………………….
   - Are We Making Progress on Our Improvement Plan? ……..
   - Are We Making Progress on the Improvement Framework?.
   - What Lessons Have We Learned So Far? …………………..

# Introduction

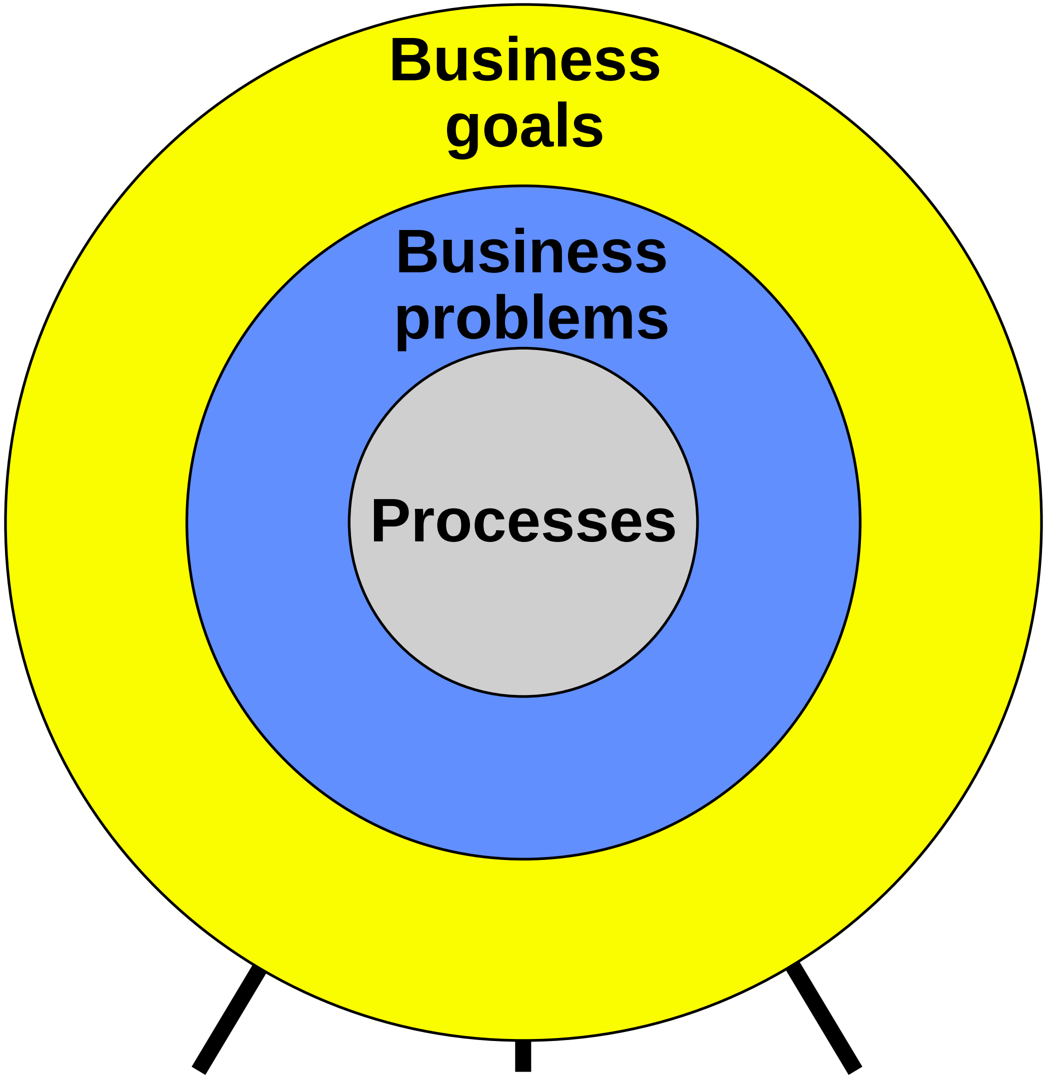
# The “Classic” Approach to PI

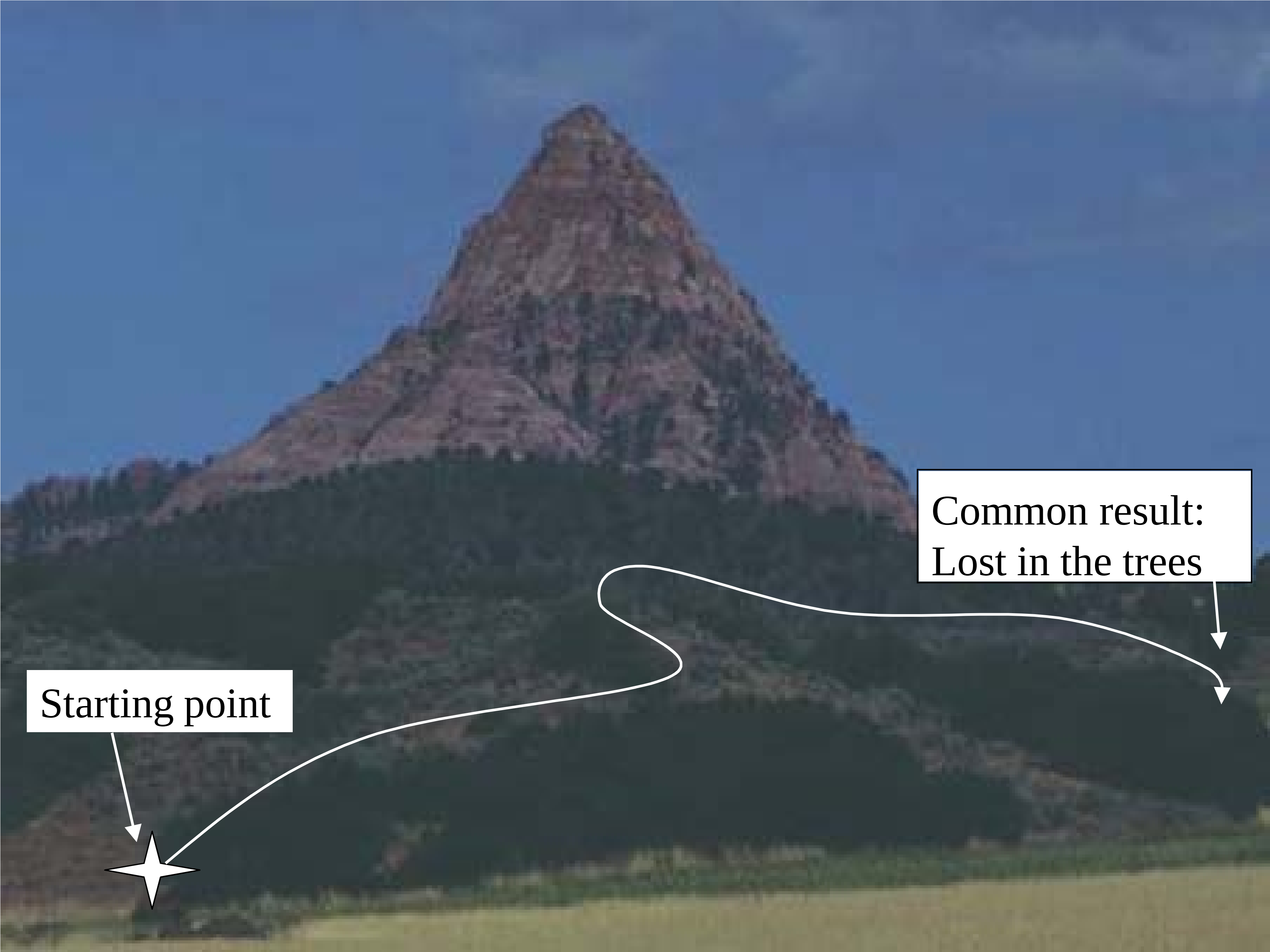
**Process-centric improvement**

- SEI CMMI
- ISO9001
- Bellcore

**It can work!**

- High risk of failure





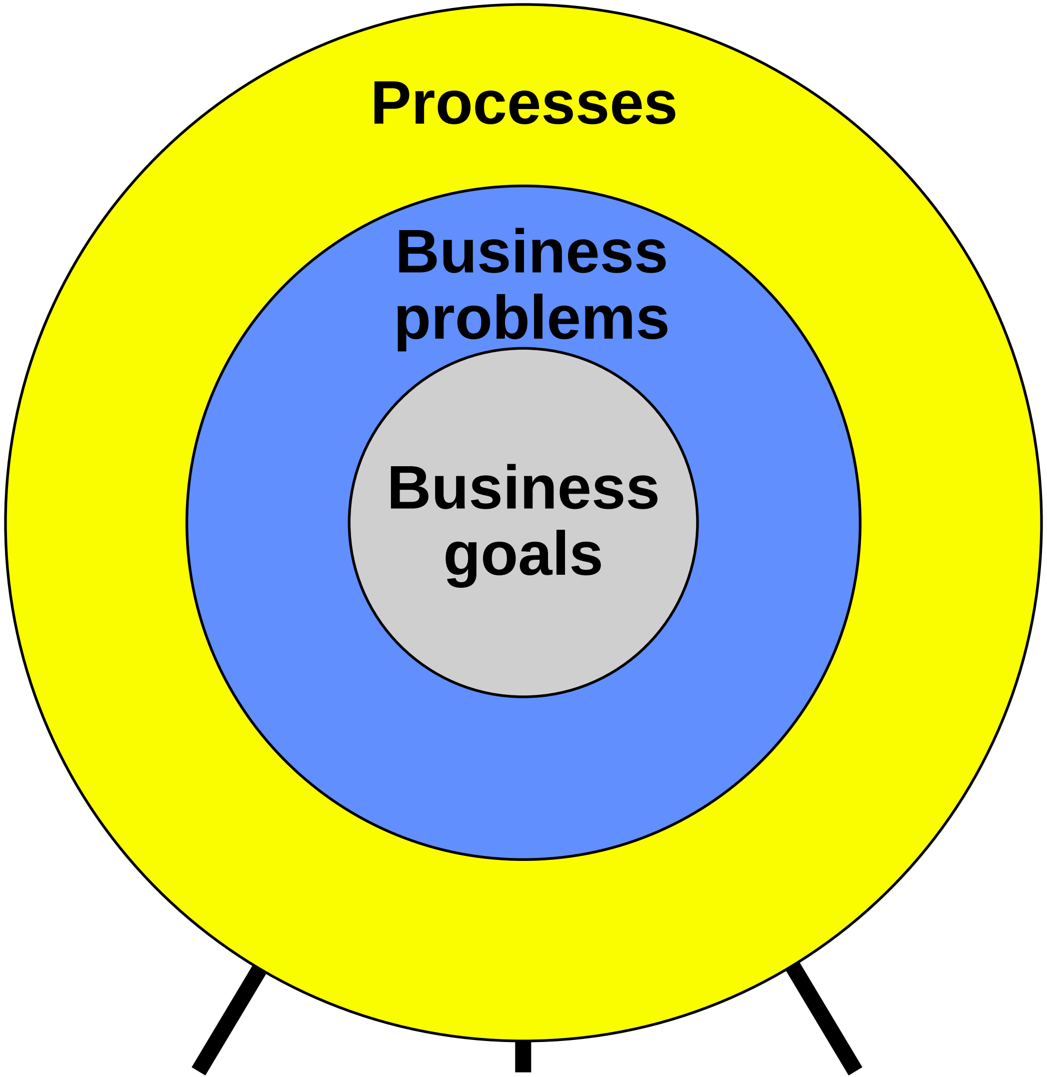
Starting point

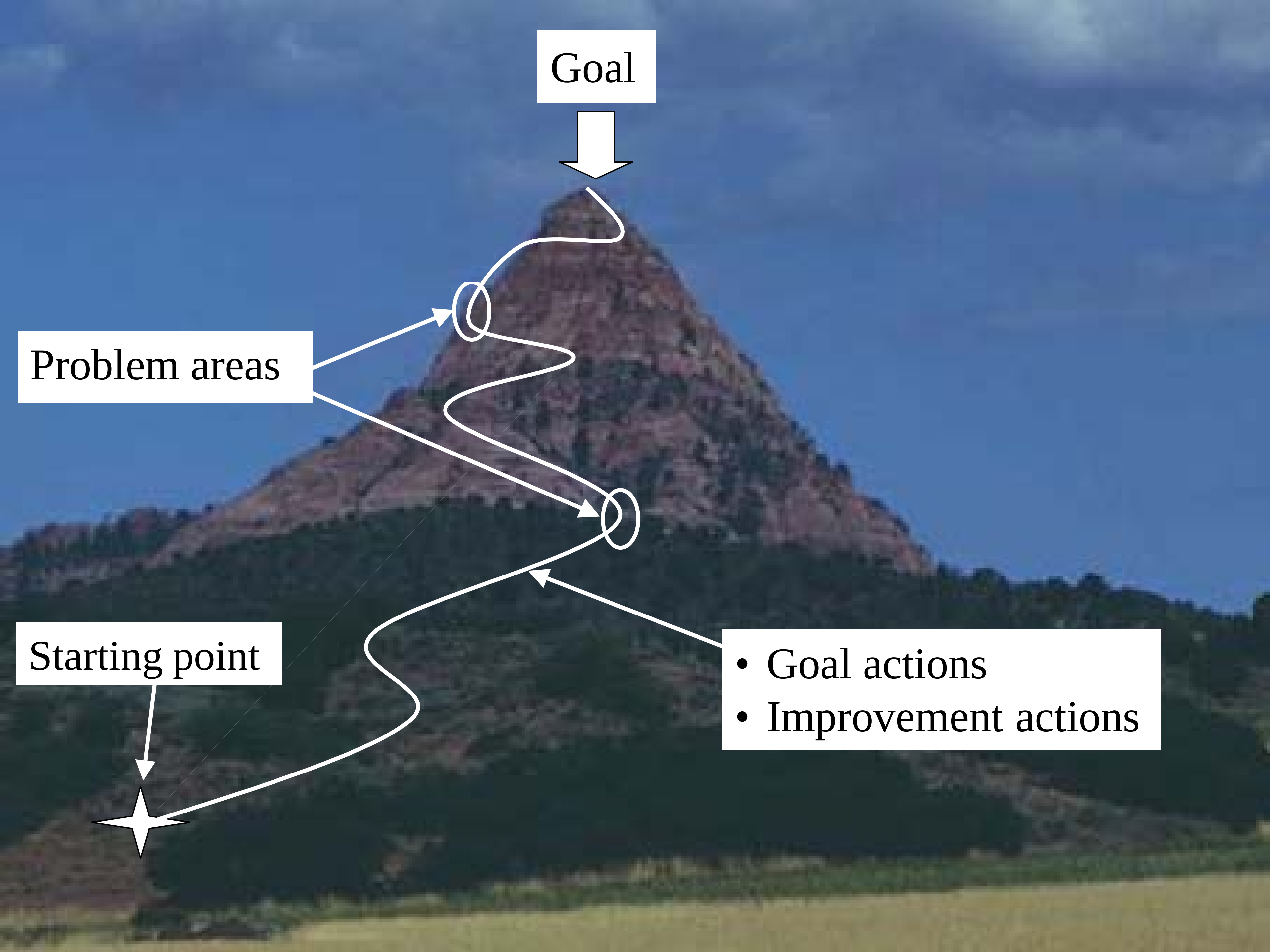
Common result:
Lost in the trees

# A Solution

Goal-problem-centric improvement

Goals and problems can be used to scope and sequence the improvement effort





Goal

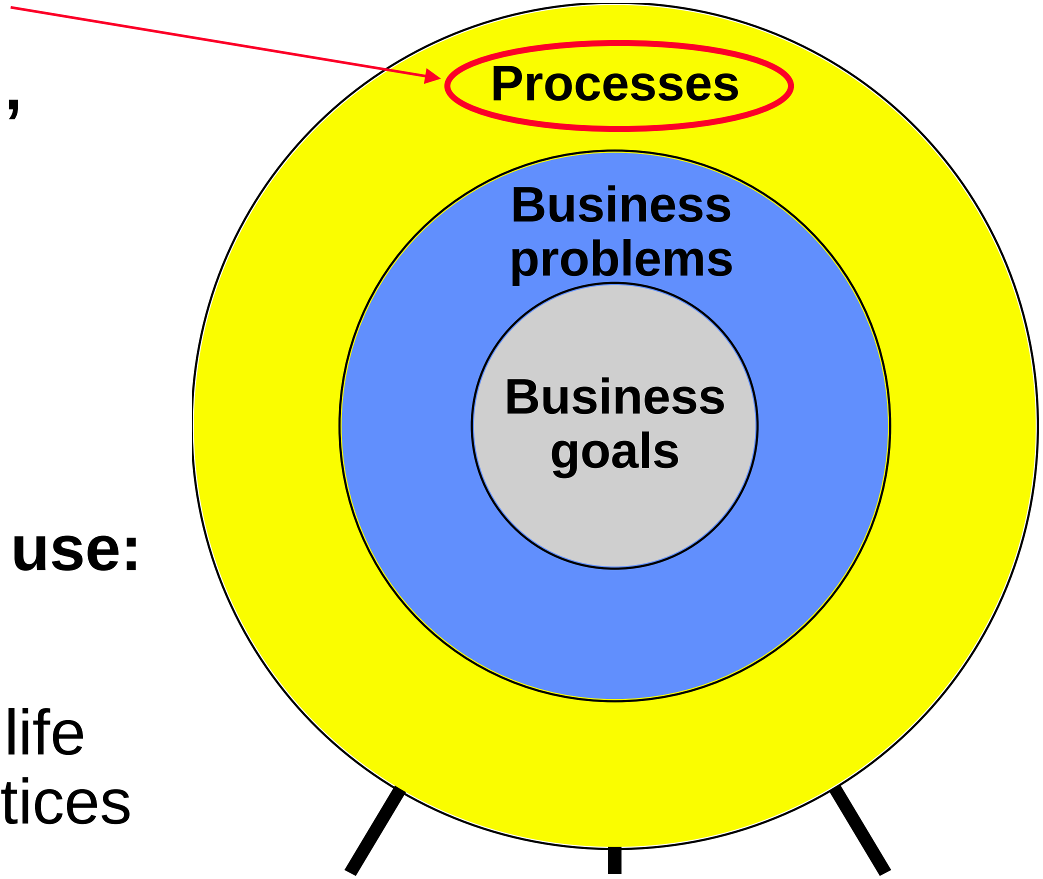
Problem areas

Starting point

- Goal actions
- Improvement actions

# Frameworks

- **Frameworks provide an optional source of improvement ideas, e.g.,**
  - – Life cycle
  - – SEI CMMI
  - – ISO9001
  - – Bellcore
- **In this workshop, either use:**
  - – No framework
  - – Current organization’s life cycle and defined practices
  - – Published framework



# Developing a Plan

"Unplanned process improvement is wishful thinking."

—Watts Humphrey, *Managing the Software Process*

# Developing a Plan

- Scope the Improvement
  1. Establish plan ownership
  2. State the major goals and problems
  3. Group the problems related to each goal
  4. Ensure that the goals and problems are crystal clear and compelling
  5. Set goal priorities
  6. Derive metrics for the goals
- **Develop an Action Plan**
- **Determine Risks and Plan to Mitigate**

# 1. Establish Plan Ownership

- **The plan meets the owner's needs, e.g.,**
  - Business goals and problems
- **The owner can be a project manager, program manager, senior manager, or division head**
- **The primary owner ≠ EPG or QA group**
  - Support functions can share ownership
- **Different individuals can be responsible for each section of the plan**

EPG = engineering process group
QA = quality assurance group

# 2. State the Major Goals and Problems - 1

## Example Goals

1. Create predictable schedules
2. Successfully deliver product X
3. Reduce rework
4. Improve the performance of our core product
5. Keep customers happy
6. Keep making a profit

# State the Major Goals and Problems - 2

## Example Problems

1. Need better requirements. Requirements tracking not in place. Changes to requirements are not tracked; code does not match specification at test time.
2. Management direction unclear for product version 2.3. Goals change often.
3. Quality department does not have training in product and test skills.
4. Unclear status of changes.
5. Lack of resources and skills allocated to design.

⋮

9. Defect repairs break essential product features.
10. Wrong files (for example, dynamic link libraries) are put on CD. Unsure of the correct ones.
11. Revising the project plan is difficult. Items drop off, new things are added, plan is out of date.
12. We don’t understand our capacity and do not have one list of all the work we have to do.
13. Schedule tracking and communication of changes to affected groups is poor.

# 3. Group the Problems Related to Each Goal - 1

- **Simplify the list by grouping the problems that prevent each goal from being achieved.**

| Goal | Problem | Problem Description |
|---|---|---|
| 1. Create predictable schedules | Problem 11 | Revising the project plan is difficult. Items drop off, new things are added, plan is out of date. |
| | Problem 12 | We don't understand our capacity and do not have one list of all the work we have to do. |
| | Problem 13 | Schedule tracking and communication of changes to affected groups is poor. |

# Group the Problems Related to Each Goal - 2

| Goal | Problem | Problem Description |
|---|---|---|
| 2. Successfully deliver product X | Problem 1 | Need better requirements. Requirements tracking not in place. Changes to requirements are not tracked; code does not match specification at test time. |
| | Problem 2 | Management direction unclear for product version 2.3. Goals change often. |

# Ensure That the Goals and Problems Are Compelling - 2

- **Example goals that are not compelling:**
  - Document all processes.
  - Develop a detailed life cycle.
  - Establish a metrics program.
- **Example goals that are more compelling:**
  - Deliver product X by Dec 15th.
  - Increase product quality to a maximum of 10 defects per release, gaining back customers X, Y, and Z, and increasing our market share by 10 percent.
  - Reduce rework to 5 percent of project effort. Use that time to create new product Y.
  - Improve schedule prediction to ± 5-day accuracy, eliminating forced cancellation of vacations.

# Ensure That the Goals and Problems Are Crystal Clear

| Original Goals | Goals Reworded for Clarity |
|---|---|
| 1. Create predictable schedules | Meet all our cost and schedule commitments |
| 2. Successfully deliver product X | Deliver product X by mm/dd/yy |
| 3. Reduce rework | Reduce rework to less than 20 percent of total project effort |
| 4. Improve the performance of our core product | Improve the performance of our core product (target to be defined) |
| 5. Keep customers happy | Achieve customer rating of 9/10 on product evaluation form |
| 6. Keep making a profit | Keep profits at 15 percent (and costs at the same level as last year) |

# Using the Approach for a Single Project

## What is your goal?

Reduce product development cycle to six to nine months for product X.

## What is preventing you from achieving the goal?

1. Changing requirements.
2. Loss of resources; difficult to replace people with specialized skills who leave the project.
3. Too many features for the six- to nine-month development cycle.
4. Poor quality of incoming code from other groups.
5. Inadequate availability of test equipment.
6. Lack of visibility within each life cycle phase. It is difficult to know whether we are ahead or behind schedule.
7. Don’t always have the resources available to complete the planned work.
8. Difficult to find defects early.

# Exercise: Scope the Improvement

1. **Form project teams**
2. **Determine the primary business goals and problems of your group**
   - Simplify the list of goals and problems by grouping the related problems under each goal
   - Verify that the scope of your improvement program is compelling
     - » If not, ask: Why do I want to achieve these goals?
3. **Discuss lessons learned**

## Result:

**What is your goal?** 3

**What is your goal?** 2

**What is your goal?** 1

Reduce product development cycle to six to nine months for product X

**What is preventing you from achieving the goal?**

1. Changing requirements
2. Loss of resources; difficult to replace people with specialized skills who leave the project
3. Too many features for the six- to nine-month development cycle
4. Poor quality of incoming code from other groups
5. Inadequate availability of test equipment
6. Lack of visibility within each life cycle phase. It is difficult to know whether we are ahead or behind schedule
7. Don Õalways have the resources available to complete the planned work
8. Difficult to find defects early

# Developing a Plan

- Scope the Improvement
- Develop an Action Plan
  1. Enumerate actions using brainstorming and a process framework
  2. Organize the action plan based on the goals and problems
  3. Add placeholders for checking progress and taking corrective action
- **Determine Risks and Plan to Mitigate**

# Develop an Action Plan

- **Develop an Action Plan**

  1. Enumerate actions using brainstorming and a process framework
     - » **1a. What actions are needed to address the problems and achieve the goals?**
     - » **1b. If a process improvement framework is being used, which elements will help the problems and goals listed?**
  2. Organize the action plan based on the goals and problems
  3. Add placeholders for checking progress and taking corrective action

# 1a. Actions for Two of the Problems - 1

| Problem | What actions are needed to address the problems and achieve the goals? |
|---|---|
| 1. Changing requirements | Baseline the requirements before design commences |
| | Only allow changes to the application interface, not to the kernel routines |
| | Improve the library control system to minimize version control errors |
| | Investigate requirements management tools |

# 1b. Framework Elements for Two of the Problems - 1

Reworded for clarity ↓

| Problem | Which elements will help the problems and goals listed? |
|---|---|
| 1. Changing requirements | Develop an understanding with the requirements providers on the meaning of the requirements. (REQM sp1.1) |
| | Assign responsibility and authority for performing the REQM process. (REQM gp2.4) |
| | Track change requests for the configuration items. (CM sp2.1) |

REQM = Requirements Management. CM = Configuration Management

# Progress on Chosen Framework -1

**95% map to Level 2**

### Example Goals

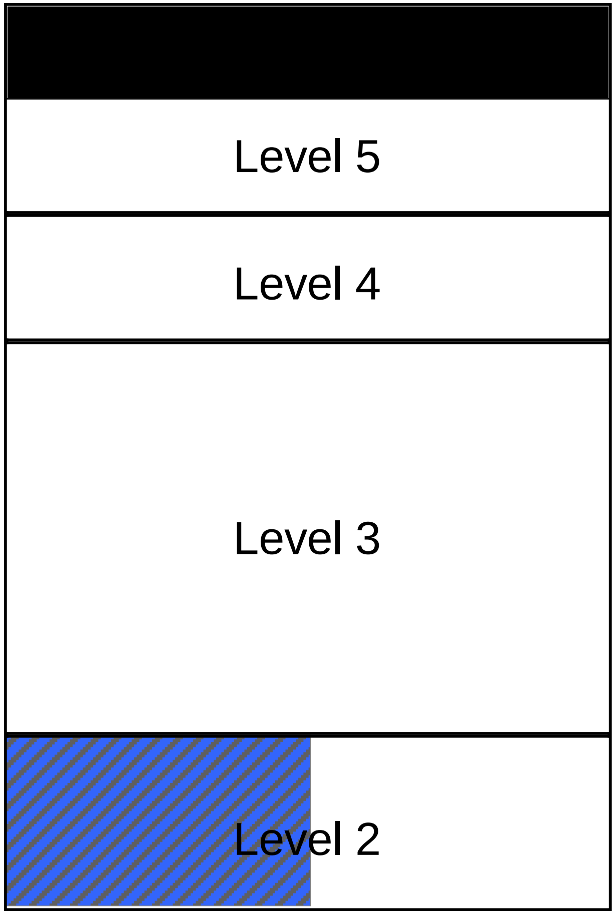
1. Create predictable schedules
2. Successfully deliver product X
3. Reduce rework
4. Improve the performance of our core product
5. Keep customers happy
6. Keep making a profit

### Example Problems

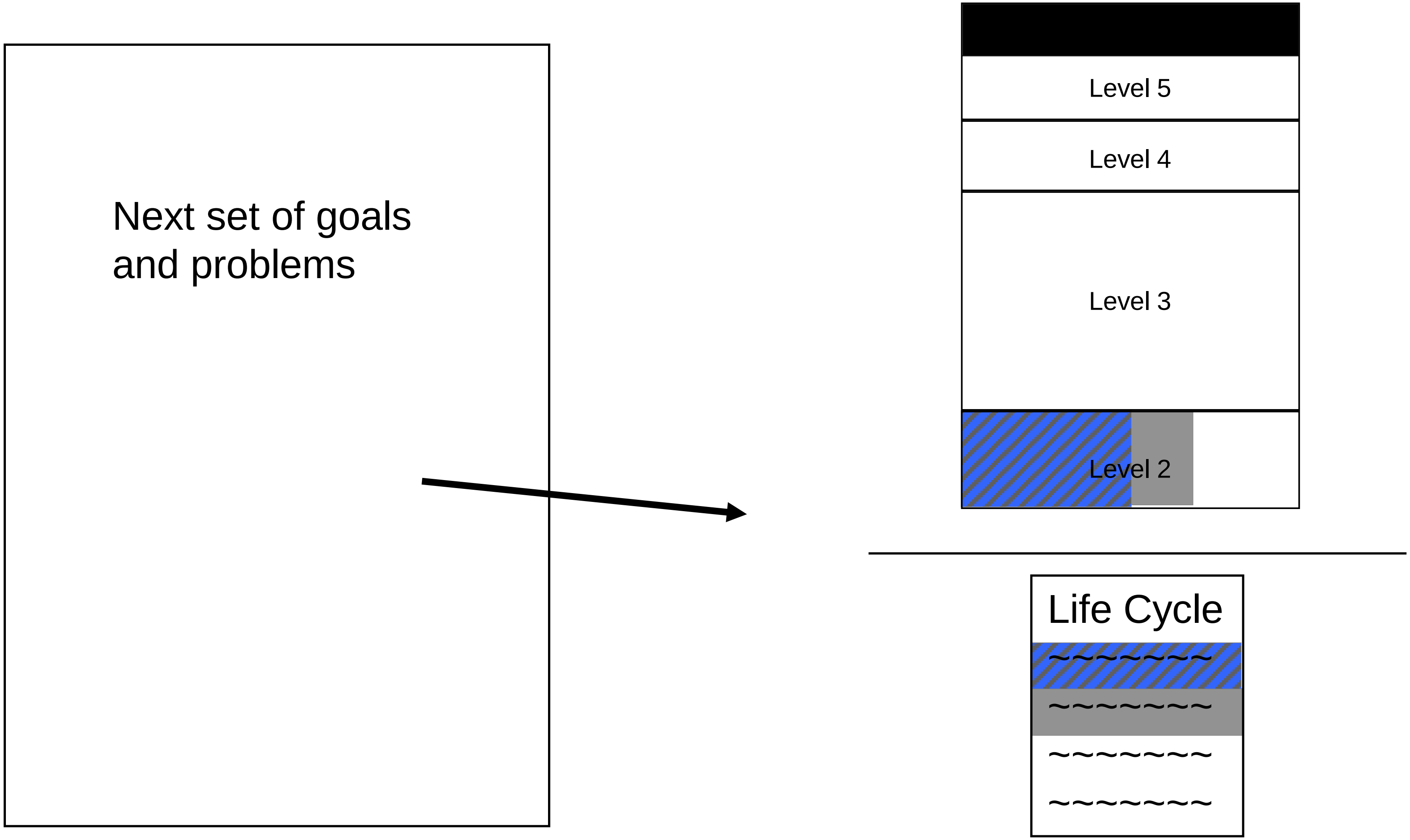
1. Need better requirements. Requirements tracking not in place. Changes to requirements are not tracked; code does not match specification at test time.
2. Management direction unclear for product version 2.3. Goals change often.
3. Quality department does not have training in product and test skills.
4. Unclear status of changes.
5. Lack of resources and skills allocated to design.

9. Defect repairs break essential product features.
10. Wrong files (for example, dynamic link libraries) are put on CD. Unsure of the correct ones.
11. Revising the project plan is difficult. Items drop off, new things are added, plan is out of date.
12. We donÕt understand our capacity and do not have one list of all the work we have to do.
13. Schedule tracking and communication of changes to affected groups is poor.

**Initial goals and problems address 43% of Level 2**

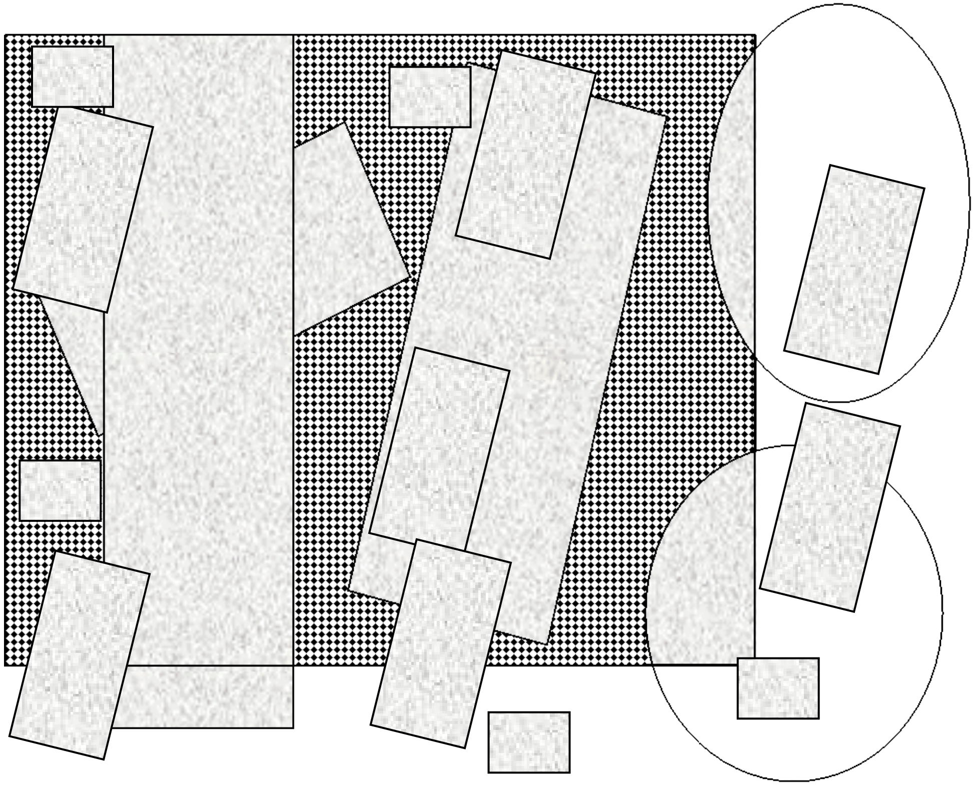


# Progress on Chosen Framework -2

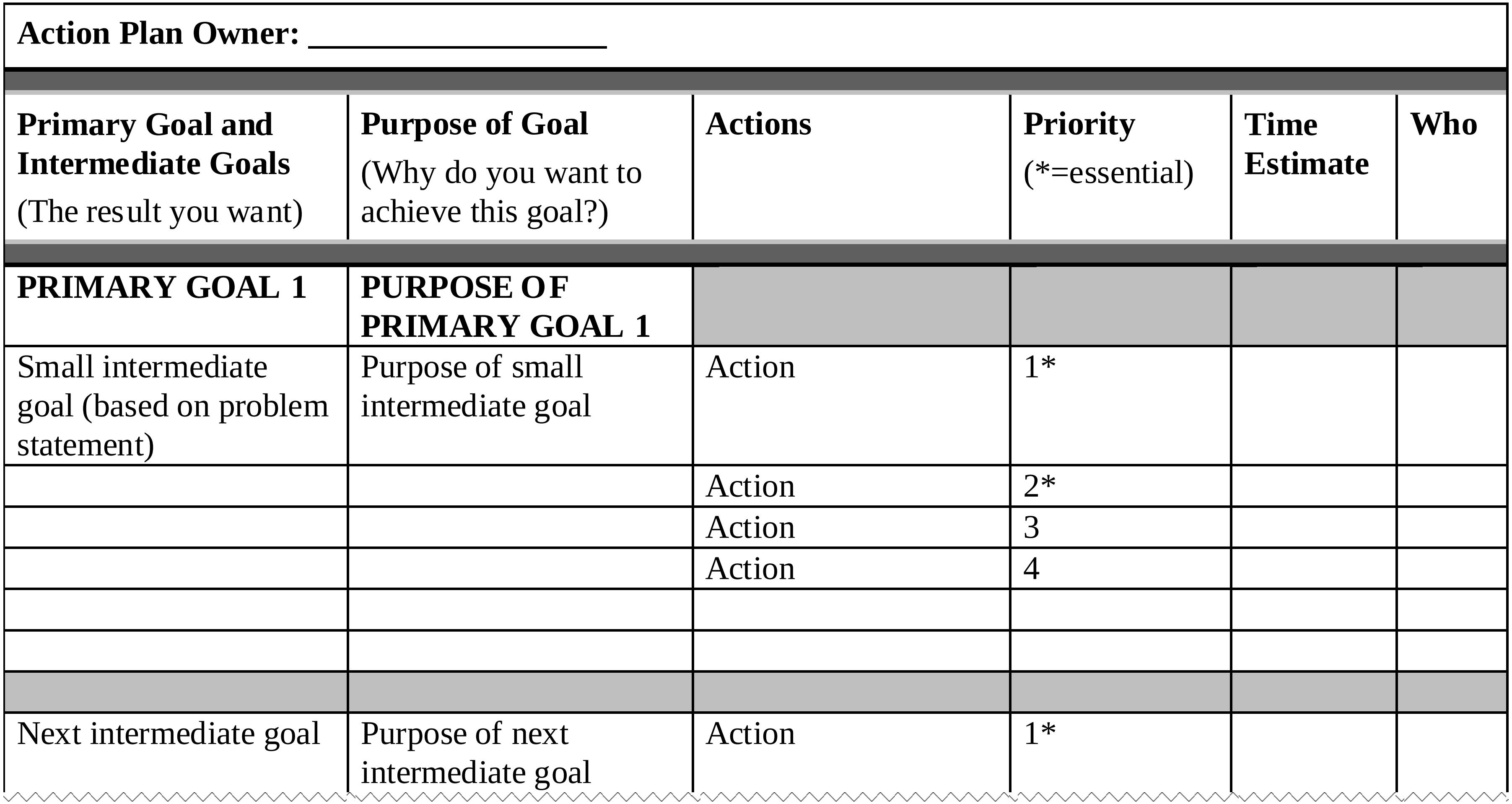

Next set of goals and problems
Level 5
Level 4
Level 3
Level 2
Life Cycle
~~~~~~~~
~~~~~~~~
~~~~~~~~
~~~~~~~~

# What to Do With the Remaining Elements?

- **Put each to good use**
  - –What problem could it solve?
- **Declare them not applicable**
  - –Check with your appraiser / auditor!
- **~~Meet the letter of the law~~**



# 2. Organize the Action Plan



| | | | | | |
|---|---|---|---|---|---|
| **Action Plan Owner: ___________________** | | | | | |
| **Primary Goal and Intermediate Goals**<br>(The result you want) | **Purpose of Goal**<br>(Why do you want to achieve this goal?) | **Actions** | **Priority**<br>(*=essential) | **Time Estimate** | **Who** |
| **PRIMARY GOAL 1** | **PURPOSE OF PRIMARY GOAL 1** | | | | |
| Small intermediate goal (based on problem statement) | Purpose of small intermediate goal | Action | 1* | | |
| | | Action | 2* | | |
| | | Action | 3 | | |
| | | Action | 4 | | |
| | | | | | |
| | | | | | |
| | | | | | |
| Next intermediate goal | Purpose of next intermediate goal | Action | 1* | | |

Template is available at www.processgroup.com/bookinfo.htm.

# Example Improvement Plan - 1

| **Primary Goal and Intermediate Goals**<br>(The results you want) | **Purpose of Goal**<br>(Why do you want to achieve the goal?) | **Actions** | **Priority**<br>(*=essential) |
|---|---|---|---|
| **Reduce product development cycle to six to nine months for product X.** | **Deliver earlier than competition.** | | |
| Manage changing requirements (based on problem 1). | Prevent schedule slips resulting from expensive scope changes. | Only allow changes to the application interface, not the kernel routines. | 1* |
| | | Assign responsibility and authority for performing the REQM process. | 2* |
| | | *Check progress and take corrective action .* | - |
| | | Improve the library control system to minimize version control errors.<br>Investigate requirements management tools. | 3 |
| | | Track change requests for the configuration items. | 4 |
| | | Develop an understanding with the requirements providers on the meaning of the requirements . | 5 |
| | | Baseline the requirements before design commences. | 6 |

Step 3: Add placeholder for checking progress and taking corrective action

# Summary - Developing a Plan

- **All improvements are tied to specific needs of the organization**
- **Goals and problems help the organization identify which pieces of an improvement framework to implement next**
- **Goals and problems establish the scope and context for each improvement**
  - When a problem has been solved or a goal addressed, a team can stop defining the process or standard
- **Practitioners and managers are motivated to work on improvement because the effort is directed toward the group’s needs**

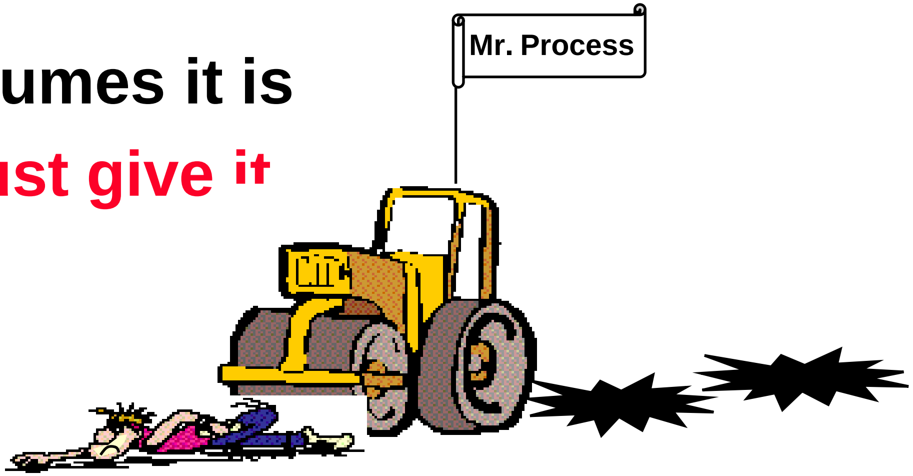
# Implementing the Plan

“Proving that the true skeptics are indeed truly skeptical achieves nothing, except that you’ve dented your pick and probably permanently diminished your credibility (and failed to appreciate the vital importance of building a fragile momentum).”

—Tom Peters, *A Passion for Excellence*

# What Too Often Happens

- A (big) process document is written
- The improvement team assumes it is done and deployment is "just give it to the people"
- The process is "deployed"

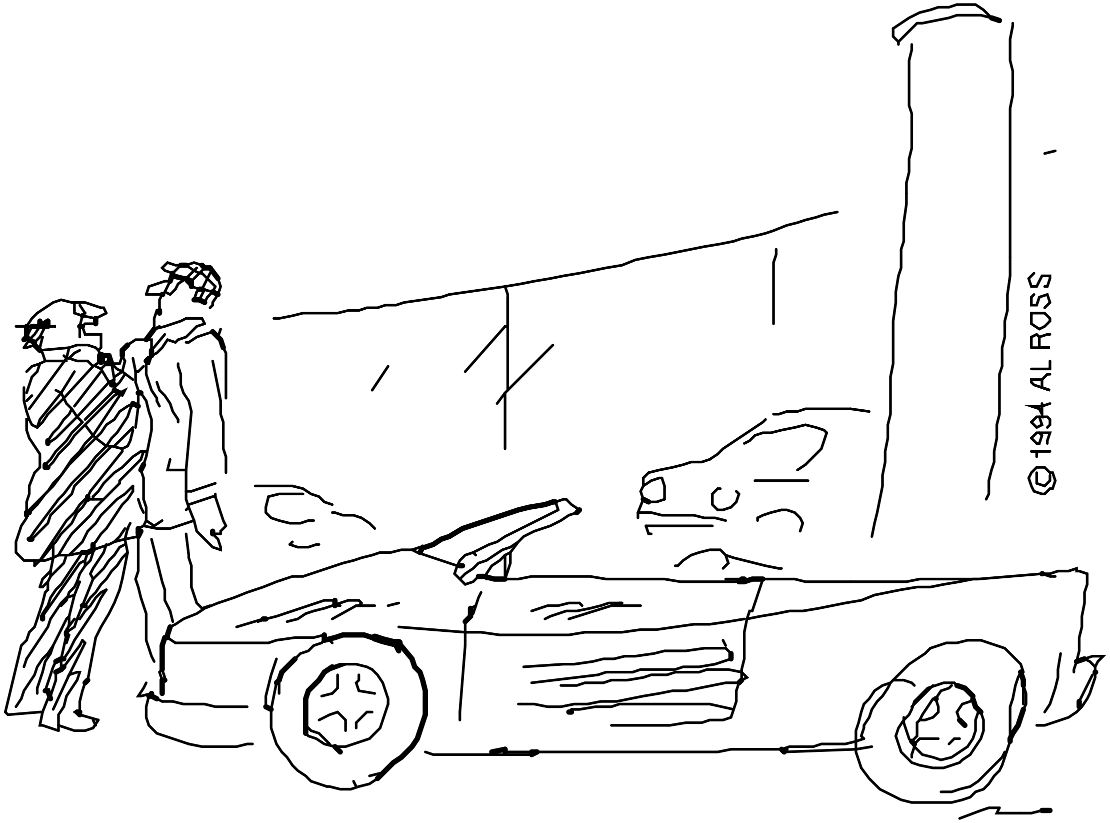


- The process is ignored, or significant resistance occurs
- The organization gives up or continues to struggle

# The Selling Aspect of Getting People to Change

- What did the sales person do in your best sales experience?

# Individuals Want to be Understood First and Then Have Their Problems Solved



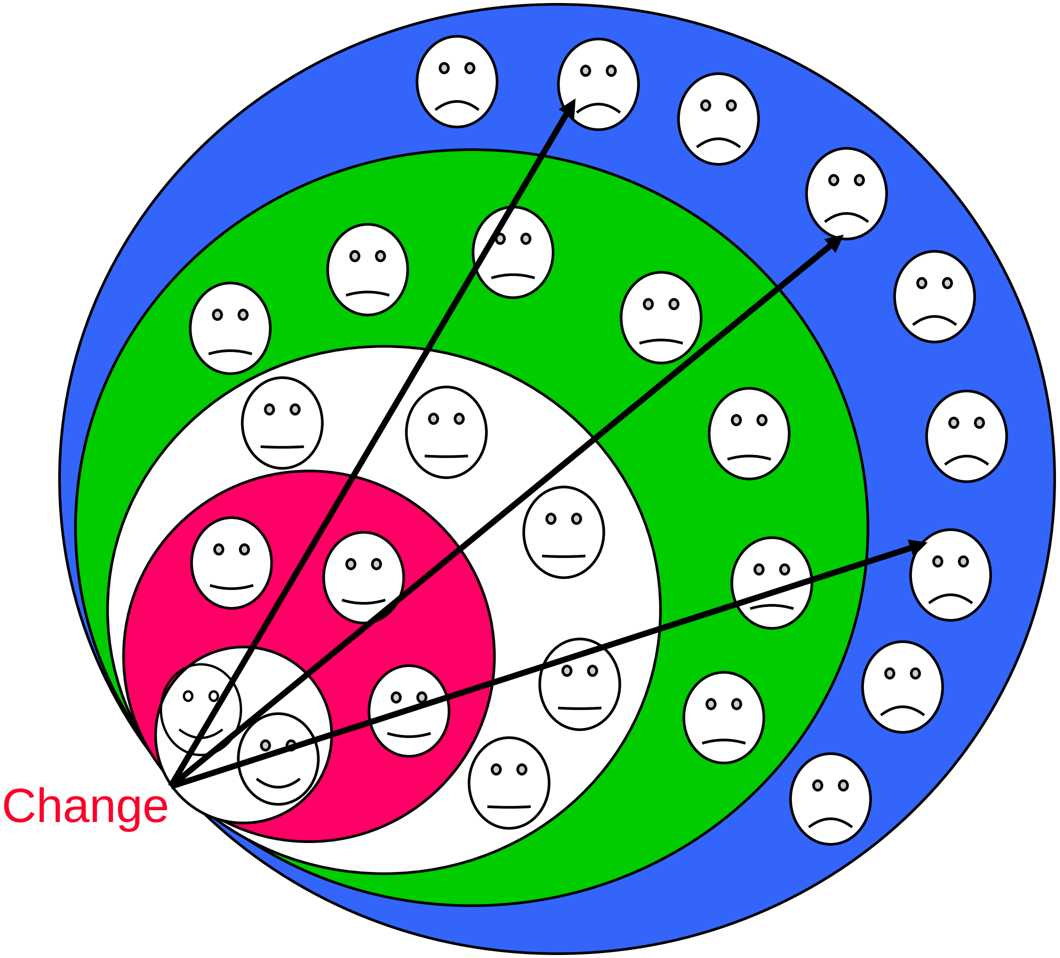
***"And I say you can afford it!"***

# How to Use *Selling*

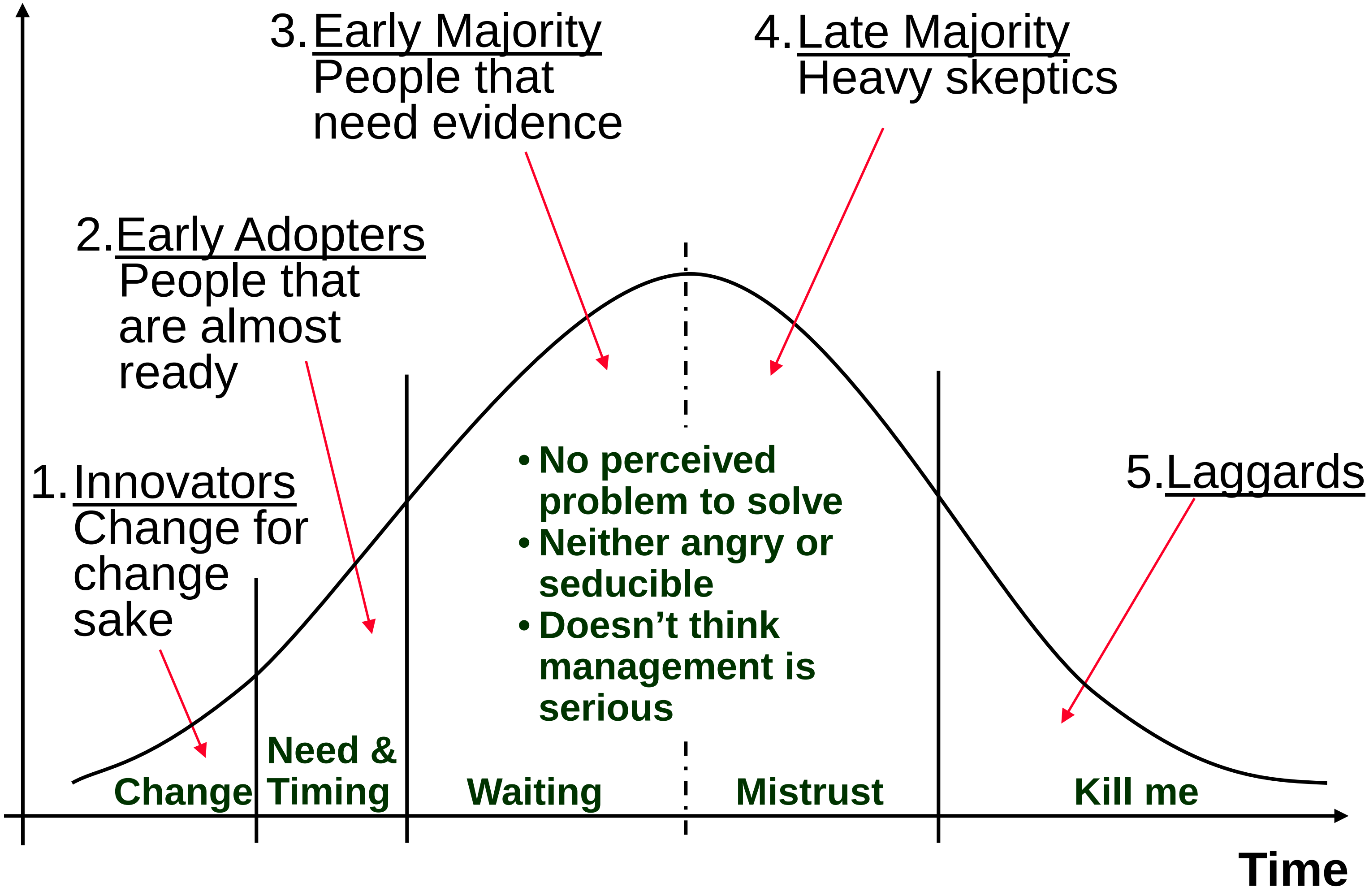
- **Forget** what you are selling
- **Understand** what the customer wants in his/her terms
  - Problems and goals
- Determine the **match** with what you have and what the customer wants
- **Solve** the customer's problem
  - may be a standard or customized solution

# Work with the Willing and Needy First

- **A planned and staged approach:**
  - Builds momentum
  - Leverages success stories
  - Provides feedback to refine the solution(s)
  - Easier to manage



# What Stages?



1. Innovators
Change for change sake

2. Early Adopters
People that are almost ready

3. Early Majority
People that need evidence

4. Late Majority
Heavy skeptics

5. Laggards

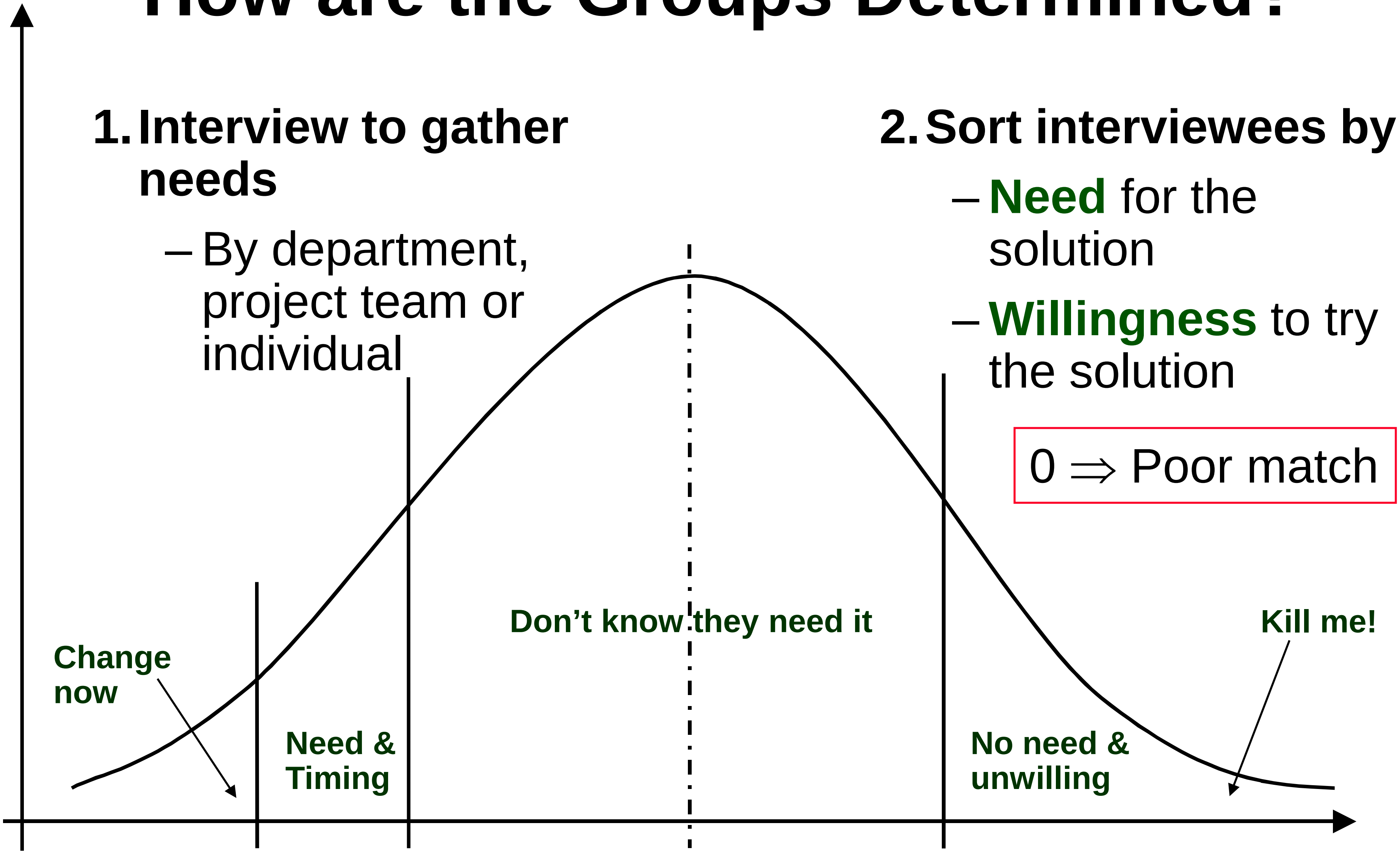
- No perceived problem to solve
- Neither angry or seducible
- Doesn't think management is serious

# How are the Groups Determined?

1. **Interview to gather needs**
   – By department, project team or individual

2. **Sort interviewees by**
   – **Need** for the solution
   – **Willingness** to try the solution

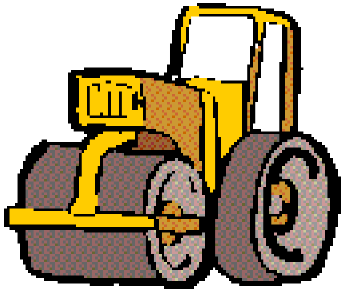
0 ⇒ Poor match



# Three Uses of the Adoption Curve

1. Increase the speed of deployment by determining with whom to work and in which order
2. Reduce the risk of failure by building and deploying the solution in increments
3. Determine when to develop a policy and issue an edict

# Summary: Implementing the Plan



- Don’t go after the hardest nut (laggard) first
- Focus on real needs (who needs what, when)
- The process provider needs to be flexible and provide appropriate, timely solutions
- PI is not *about* documentation
- Management can lead

# Checking Progress

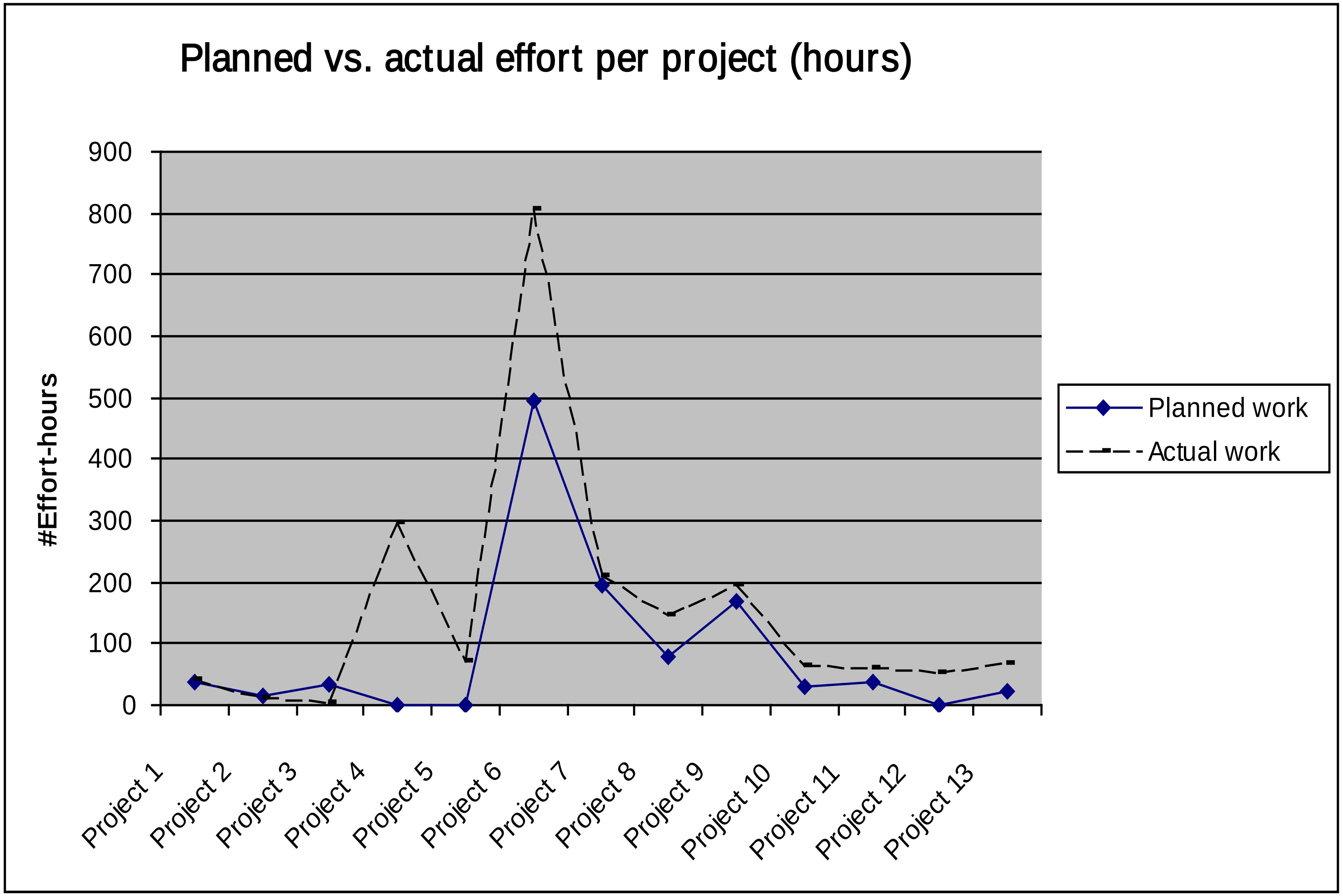
"You can design a measurement system for any conclusion you wish to draw."

—Gerald Weinberg, *Quality Software Management*

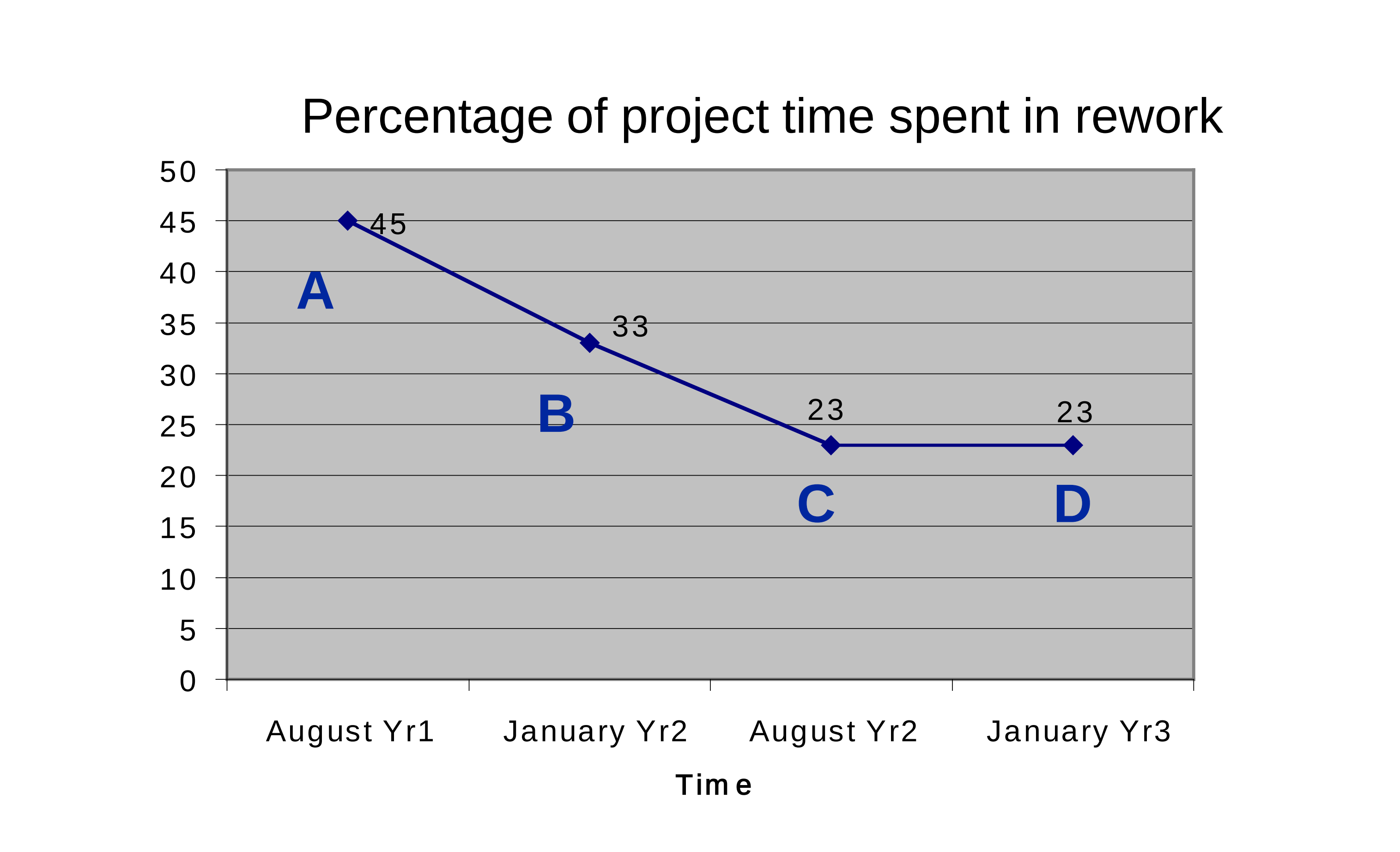
# Checking Progress

- – Are We Making Progress on the Goals?
- – Are We Making Progress on Our Improvement Plan?
- – Are We Making Progress on the Improvement Framework?
- – What Lessons Have We Learned So Far?

# Goal: Meet all Our Cost and Schedule Commitments

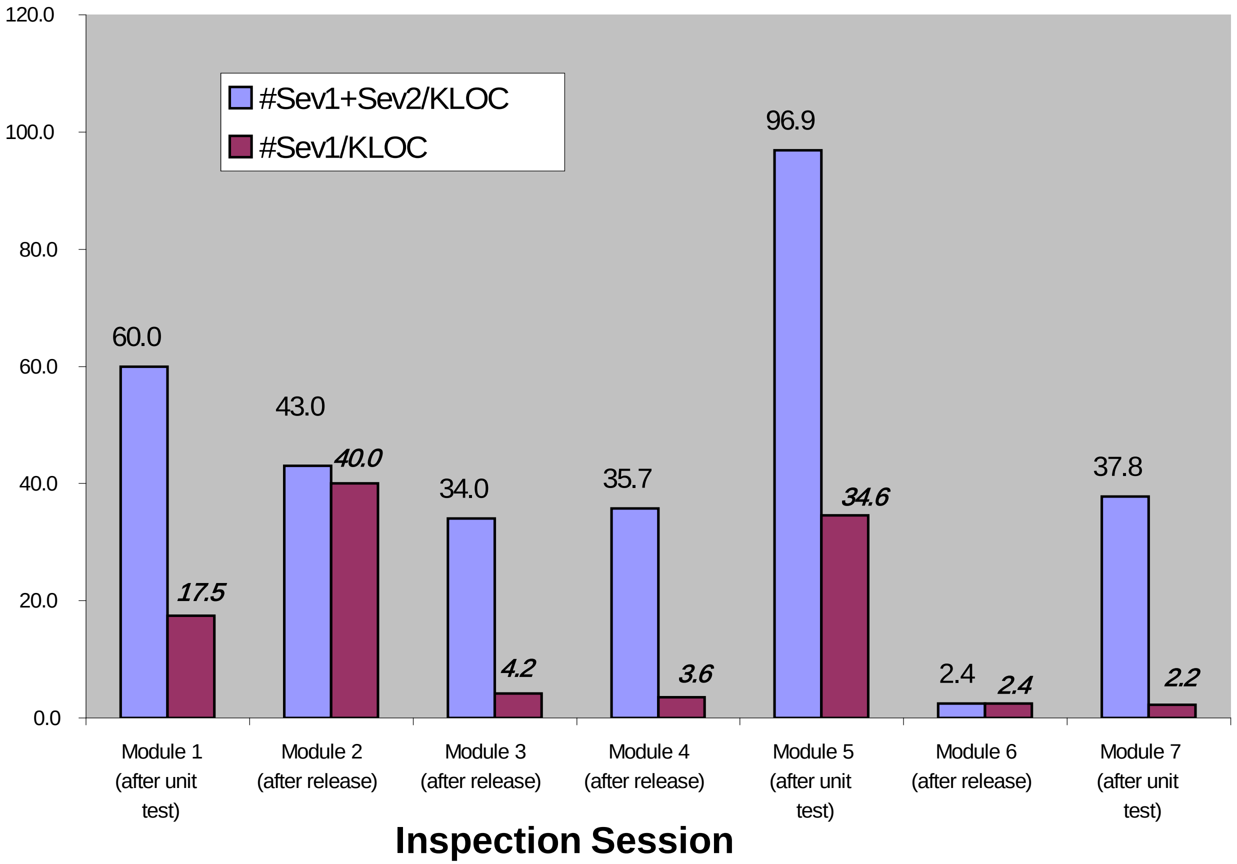


# Goal: Reduce Rework to Less Than 20 Percent of Total Project Effort - 1

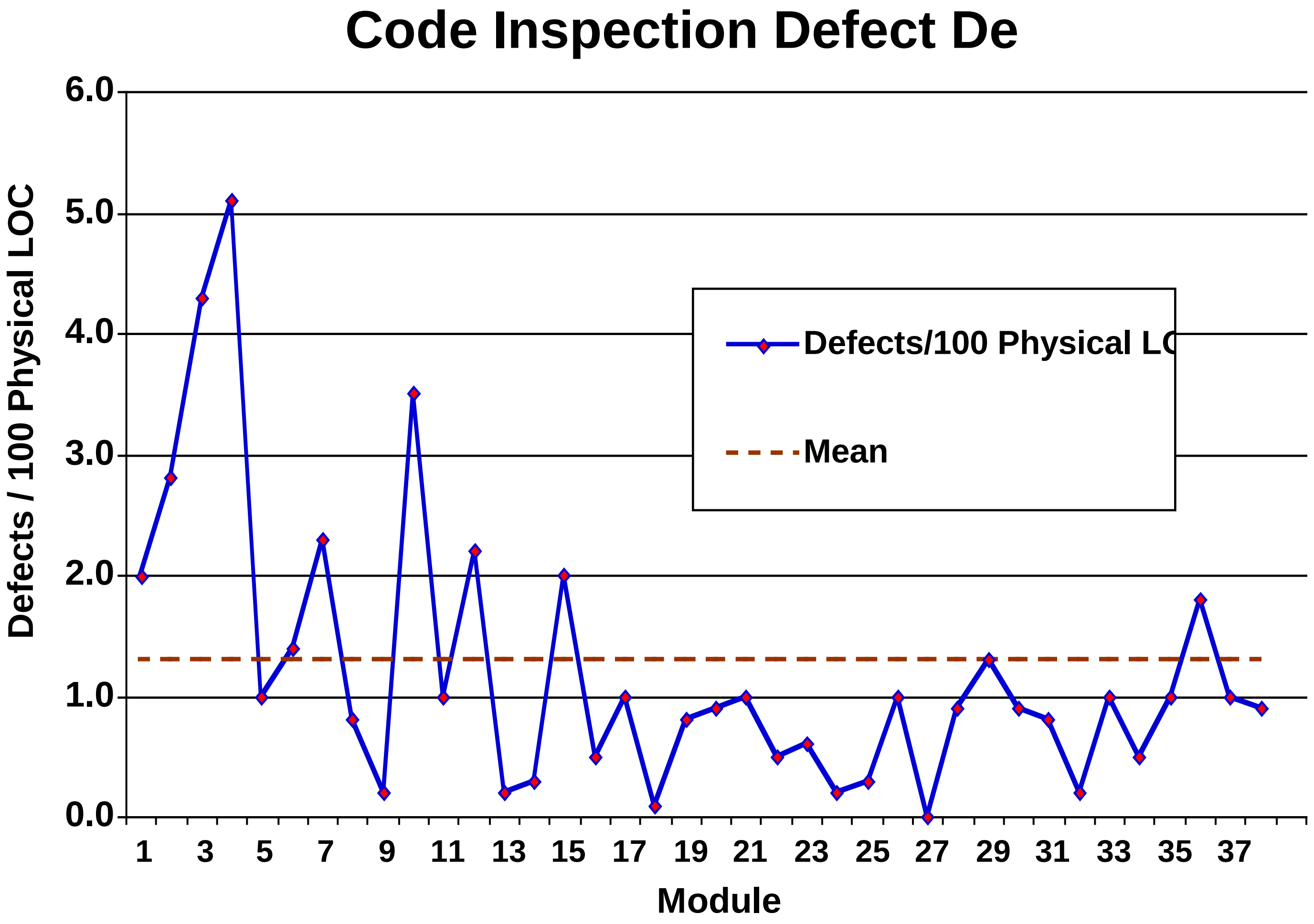


# Goal: Reduce Rework to Less Than 20 Percent of Total Project Effort - 2

Java/C++ Inspections – Severity 1 + Severity 2 Defects per Thousands of Lines of Code

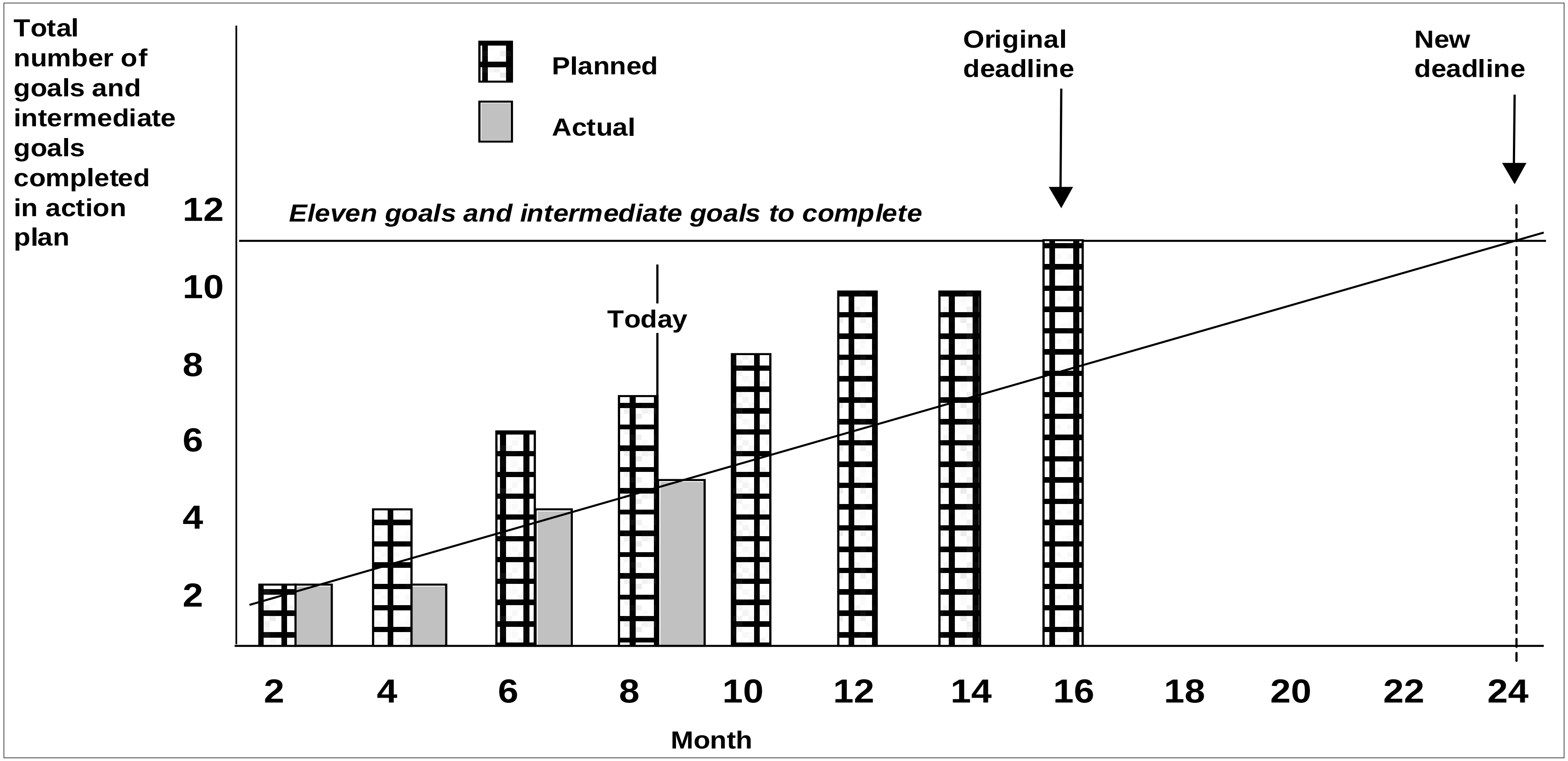


# Goal: Reduce Rework to Less Than 20 Percent of Total Project Effort - 3



- Manufacturing control system
- OO/C++
- 167KLOC
- 13 defects/KLOC in code
- 1.38 defects/KLOC in test

# Are we Making Progress on Our Improvement Plan?



Trend diagram tracking goal and intermediate goal completion

# Are We Making Progress on the Improvement Framework? - 1

## Method 1: Count actions that are from the framework

| **Primary Goal and Intermediate Goals**<br>(The results you want) | **Purpose of Goal**<br>(Why do you want to achieve the goal?) | **Actions** | **Priority**<br>(*=essential) |
|---|---|---|---|
| **Reduce product development cycle to six to nine months for product X.** | **Deliver earlier than competition.** | | |
| Manage changing requirements (based on problem 1). | Prevent schedule slips resulting from expensive scope changes. | Only allow changes to the application interface, not the kernel routines. | 1* |
| | | Assign responsibility and authority for performing the REQM process. | 2* ✓ |
| | | *Check progress and take corrective action .* | - |
| | | Improve the library control system to minimize version control errors.<br>Investigate requirements management tools. | 3 |
| | | Track change requests for the configuration items. | 4 ✓ |
| | | Develop an understanding with the requirements providers on the meaning of the requirements . | 5 ✓ |
| | | Baseline the requirements before design commences. | 6 |

# Are We Making Progress on the Improvement Framework? - 2

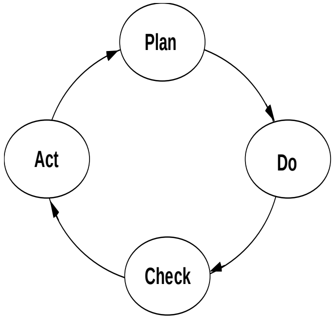
## Method 2: Conduct a mini-assessment to establish adoption of practices*

**Purpose:**

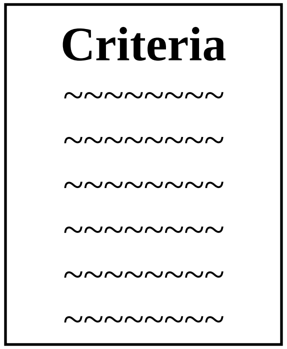
- To **evaluate improvement progress** and make necessary adjustments

**Method:**

- Develop a **checklist for a verbal interview with each project**
- Conduct interviews with each project (2-3 times per year)

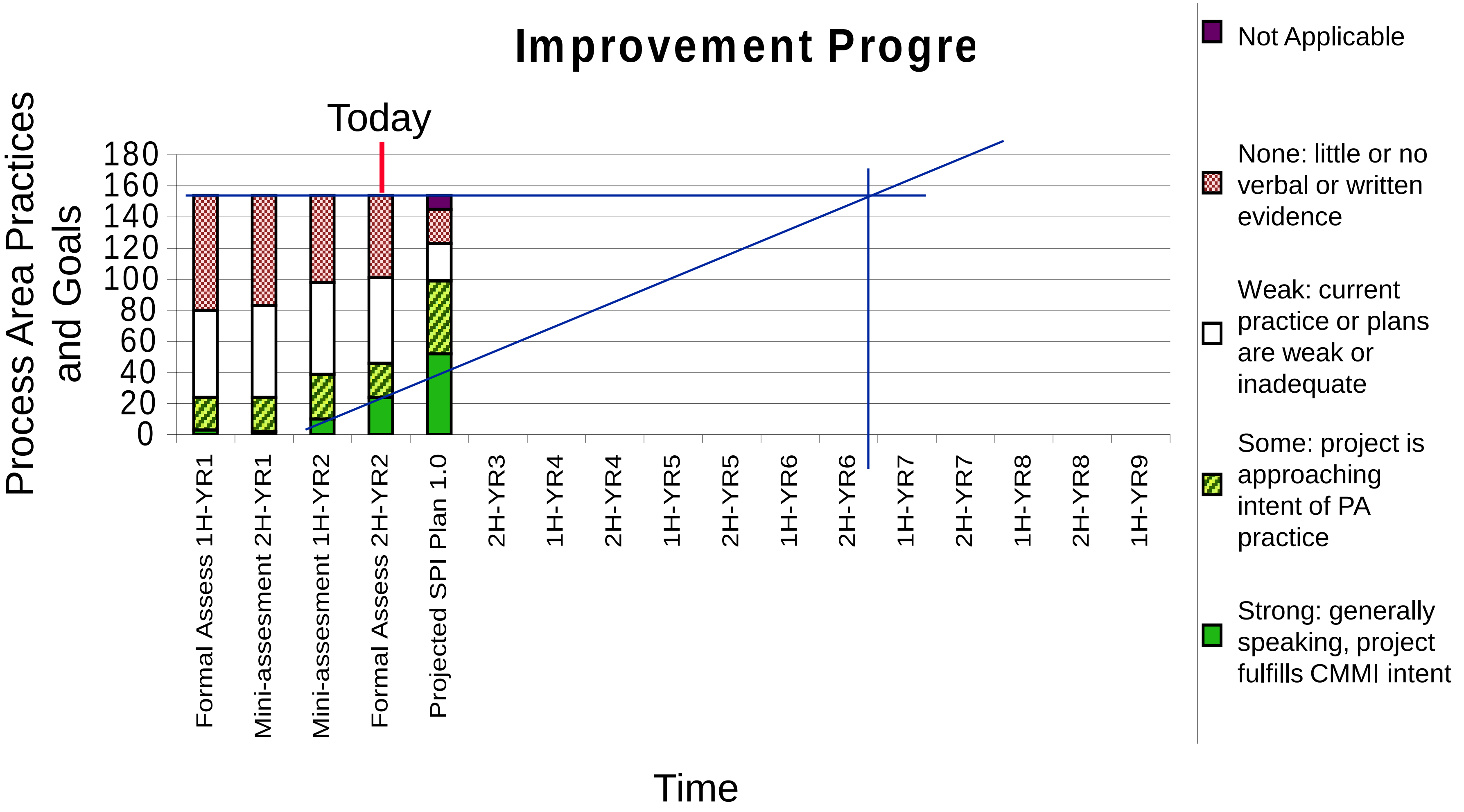


Mini-assessment

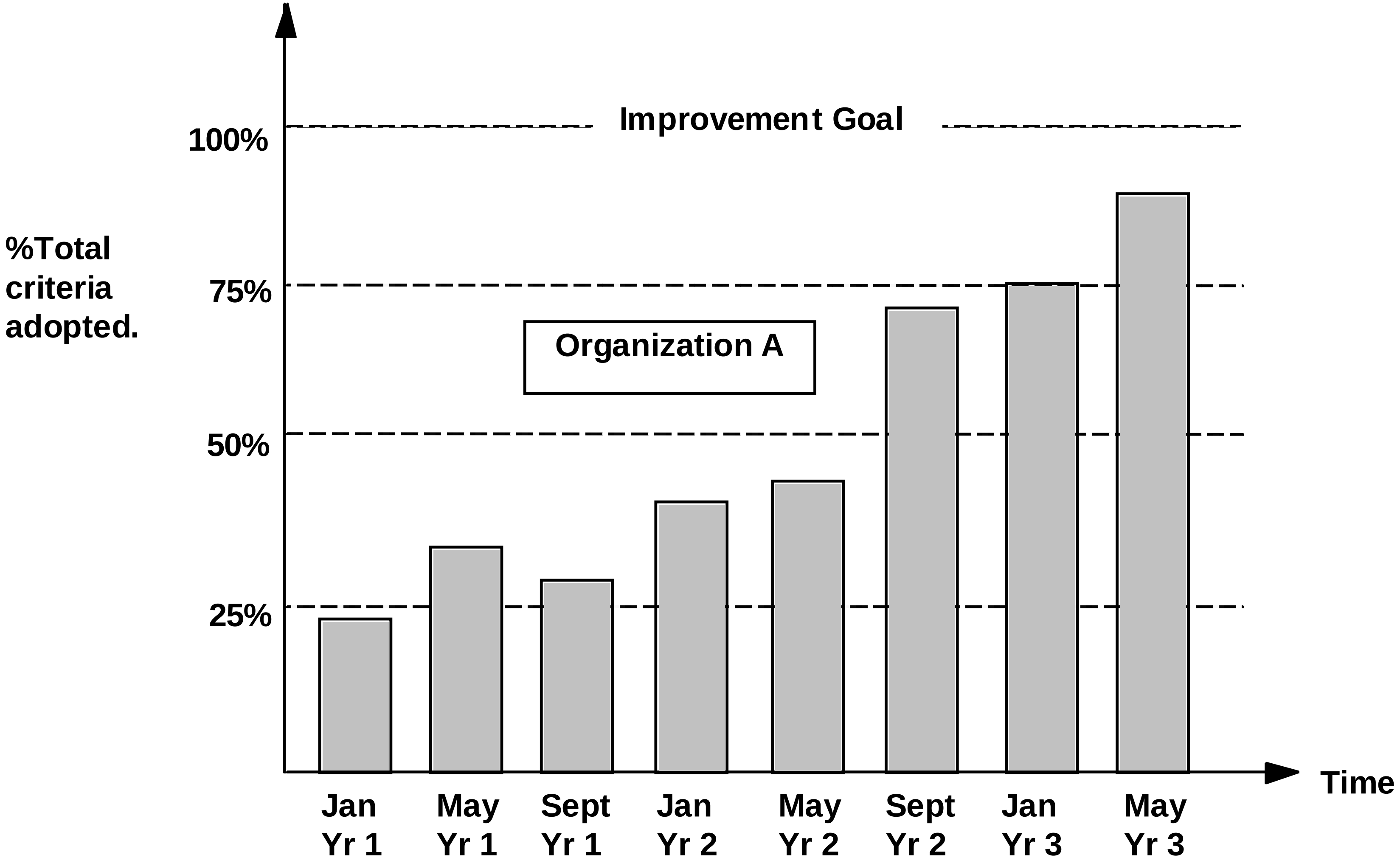


*Potter, N., Sakry, M., "Making Process Improvement Work - A Concise Action Guide for Software Managers and Practitioners," Appendix F. Addison-Wesley, 2002.

# Example Mini-assessment Data - 1

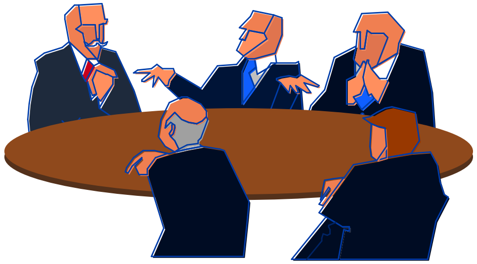


# Example Mini-assessment Data - 2



# What Lessons Have we Learned so Far?

- **Invite people who are willing to be frank and candid**
  - e.g., PI users, skeptics, managers
- **Select a good objective facilitator**
- **Two hours or less to avoid team fatigue**



## Lessons learned agenda

1. **Clarify the scope of the session** **[10 mins]**
2. **Determine strengths (what went well)** **[20 mins]**
3. **Determine areas for improvement** **[30 mins]**
4. **Set priorities** **[30 mins]**
5. **Determine corrective actions** **[30 mins]**
   1. Where to use the lesson
   2. Specific corrective actions

# Lessons Learned - Strengths

| Lesson | Where to Use Lesson |
|---|---|
| **Decentralizing the action plan gives each project team ownership over its plan.**<br>Corrective action (CA) = Continue having three separate action plans, one for each of the three product lines. | **Planning** |
| **Don’t preach when an example can say everything for you.**<br>CA = Have one project each month conduct a one-hour briefing describing the use and benefits of a new technique. | **Implementing** |
| **Guide people in applying each new technique to their work. People have so much going on that they do not know where to start.**<br>CA = For each process in the process assets library (PAL), add tailoring guidelines to explain when the process should be used. Provide one-on-one coaching to new project teams. | **Implementing** |

# Lessons Learned - Improvement Areas

| | |
|---|---|
| **The process-centric approach was very difficult to sell.**<br>CA = adopt the goal-problem approach. | Planning |
| **Using the same communication technique as everyone else allows the message to be lost.**<br>CA = use bright pink 8.5 x 11-inch cards & pizza lunches. | Implementing |
| **Allowing private data to become public sets perilous expectations.**<br>CA = brief management on new metrics policy. | Planning |
| **Be careful of what information you ask for! [Process Assets Library]**<br>CA = stop measuring the % of projects that submit to the PAL. Clean out the PAL. | Planning |
| **Using a scoring system for process adoption can encourage inappropriate behavior.**<br>CA = stop measuring #inspections/year. Re-look at all metrics that can be optimized but lead to little benefit. | Checking |

# Summary - Checking Progress

- Measure what you care about
- Practice measuring
- Lessons-learned data provides additional feedback
- Take corrective action based on what you learn

# References

1. Basili, V., and D. Weiss. “A Methodology for Collecting Valid Software Engineering Data.” IEEE Transactions on Software Engineering 1984;SE-10(6):728–738.

2. Block, P., “Flawless Consulting: A Guide to Getting Your Expertise Used.” 2nd ed. San Francisco: Jossey-Bass/Pfeiffer, 1999.

3. 1.1: CMMI Product Development Team. CMMI for Systems Engineering/Software Engineering/Integrated Product and Process Development. Version 1.1 (CMMI-SE/SW/IPPD, v1.1), staged representation. CMU/SEI-2002-TR-004, ESC-TR-2002-004, Pittsburgh: SEI, November 2002.

4. 1.2: CMMI Product Development Team. CMMI for Development. Version 1.2 (CMMI-DEV, v1.2). CMU/SEI-2006-TR-008CMU/SEI-2006-TR-008, ESC-TR-2006-008. August 2006.

5. Dion, R., “Process Improvement and the Corporate Balance Sheet.” IEEE Software 1993:10, no. 4:28–35.

6. Florac, W., and A. Carleton. Measuring the Software Process: Statistical Process Control for Software Process Improvement. Reading, MA: Addison-Wesley, 1999.

7. Grady, R., and D. Caswell. Software Metrics: Establishing a Company-wide Program. Englewood Cliffs, NJ: Prentice-Hall, 1987.

8. Grady, R., Practical Software Metrics for Project Management and Process Improvement. Englewood Cliffs, NJ: Prentice-Hall, 1992.

10. Humphrey, W., “Software Quality Assurance.” In: Managing the Software Process. Reading, MA: Addison-Wesley,1989:137–153.

# References

11. Humphrey, W., A Discipline for Software Engineering. Reading, MA: Addison-Wesley, 1995.
12. Moore, G., Crossing the Chasm. New York: Harper-Business, 1991.
13. Carnegie Mellon University/Software Engineering Institute. Edited by: M. Paulk, C. Weber, B. Curtis and M. B. Chrissis. The Capability Maturity Model: Guidelines for Improving the Software Process. Reading, MA: Addison-Wesley, 1995.
14. Robbins A., The Time of Your Life. Audiocassette program. San Diego: Robbins Research International, 1998.
15. Rogers, E., Diffusion of Innovations. New York: The Free Press, 1962.
16. Potter, N., and M. Sakry. “Practical CMM.” Software Development 2001;9:65–69.
17. Potter, N., Sakry, M., “Making Process Improvement Work - A Concise Action Guide for Software Managers and Practitioners,” Addison-Wesley, 2002.
18. Weinberg, G., The Secrets of Consulting: A Guide to Giving and Getting Advice Successfully. New York: Dorset House Publishing, 1985.
19. Wiegers, K. Software Requirements. Redmond, WA: Microsoft Press, 1999.
20. Yamamura, G., and G. Wigle. “SEI CMM Level 5: For the Right Reasons.” CROSSTALK—The Journal of Defense Software Engineering. 1997;10,8:3–8.
21. CMMI: http://www.sei.cmu.edu/cmmi/models/models.html
22. ROI information: http://www.processgroup.com/resources.htm (see ROI Data)