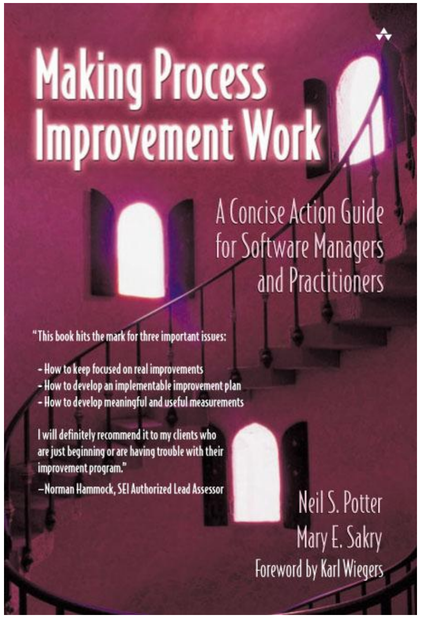
Making Process Improvement Work

A Concise Action Guide for Software Managers and Practitioners

Neil Potter

Mary Sakry

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



"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

Agenda - 1

Page

1. **Introduction…………………………………………….... 3**
2. **Developing a Plan……………………………………. 7**
   - Scope the Improvement………………………………... 7
   - Exercise…………………………………………….…. 16
   - Develop an Action Plan……………………………….. 16
   - Exercise………………………………………………. 24
   - Determine Risks and Plan to Mitigate…………….…… 24
   - Exercise………………………………………………. 30

Agenda - 2

Page

3. Implementing the Plan………………………………. 31

- Sell Solutions Based on Needs………………………………. 32
- Work with the Willing and Needy First……………………….. 34
- Keep Focused on the Goals and Problems…………………. 45
- Align the Behaviors of Managers and Practitioners………… 48
- Avoiding a Documentation Glut………………………………. 50
- Exercise ………………………………………………………. 58

4. Checking Progress……………………………………. 59

- Are We Making Progress on the Goals? ……………………. 61
- Are We Making Progress on Our Improvement Plan? …….. 65
- Are We Making Progress on the Improvement Framework?. 67
- What Lessons Have We Learned So Far? ………………….. 73
- Exercise…………………………………………………….…. 76

Agenda - 3

Page

5. Appendix……………………………………………..77

- – Engineering Process Group (EPG)……………………………… 77
- – References…………………………………………………. 82
- – Management Overview……………………………………. --

Your Expectations

- **Your job**
- **Experiences with process improvement and frameworks**
  - (e.g., CMM, CMMI, ISO9001)
- **Expectations for this class (5/5 score?)**

Introduction

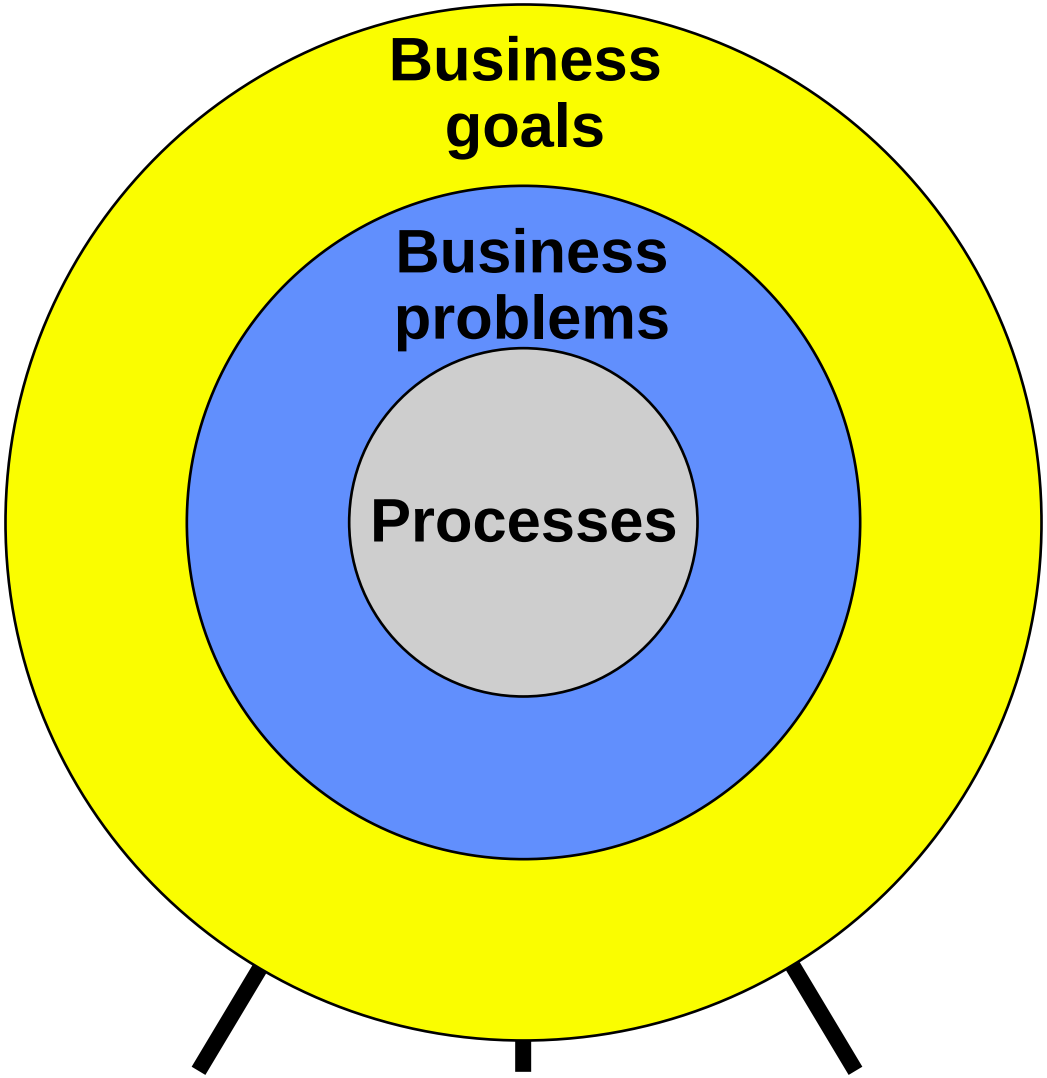
The "Classic" Approach to PI

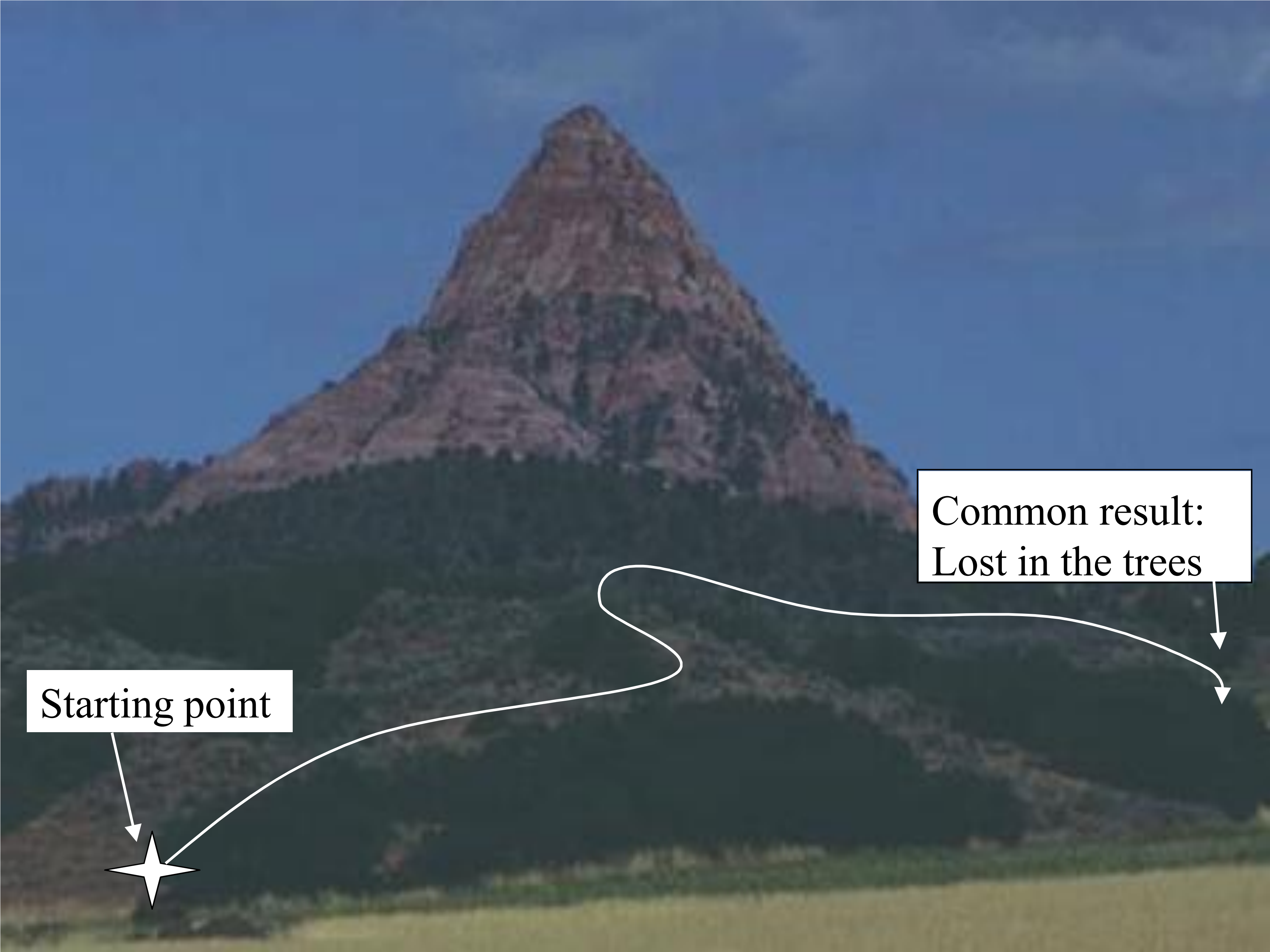
Process-centric improvement

- SEI CMMI
- ISO9001
- Bellcore

It can work!

- High risk of failure





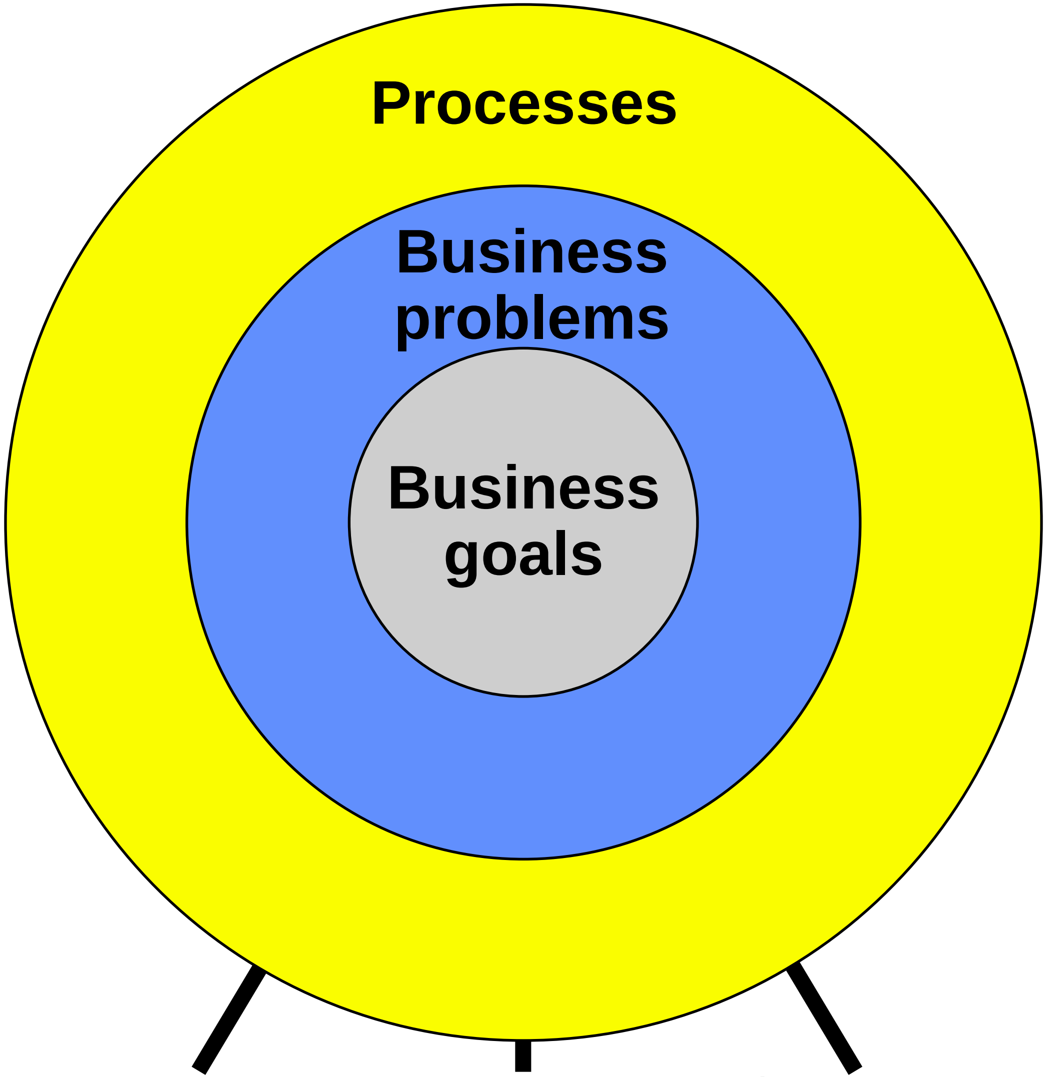
Common result: Lost in the trees

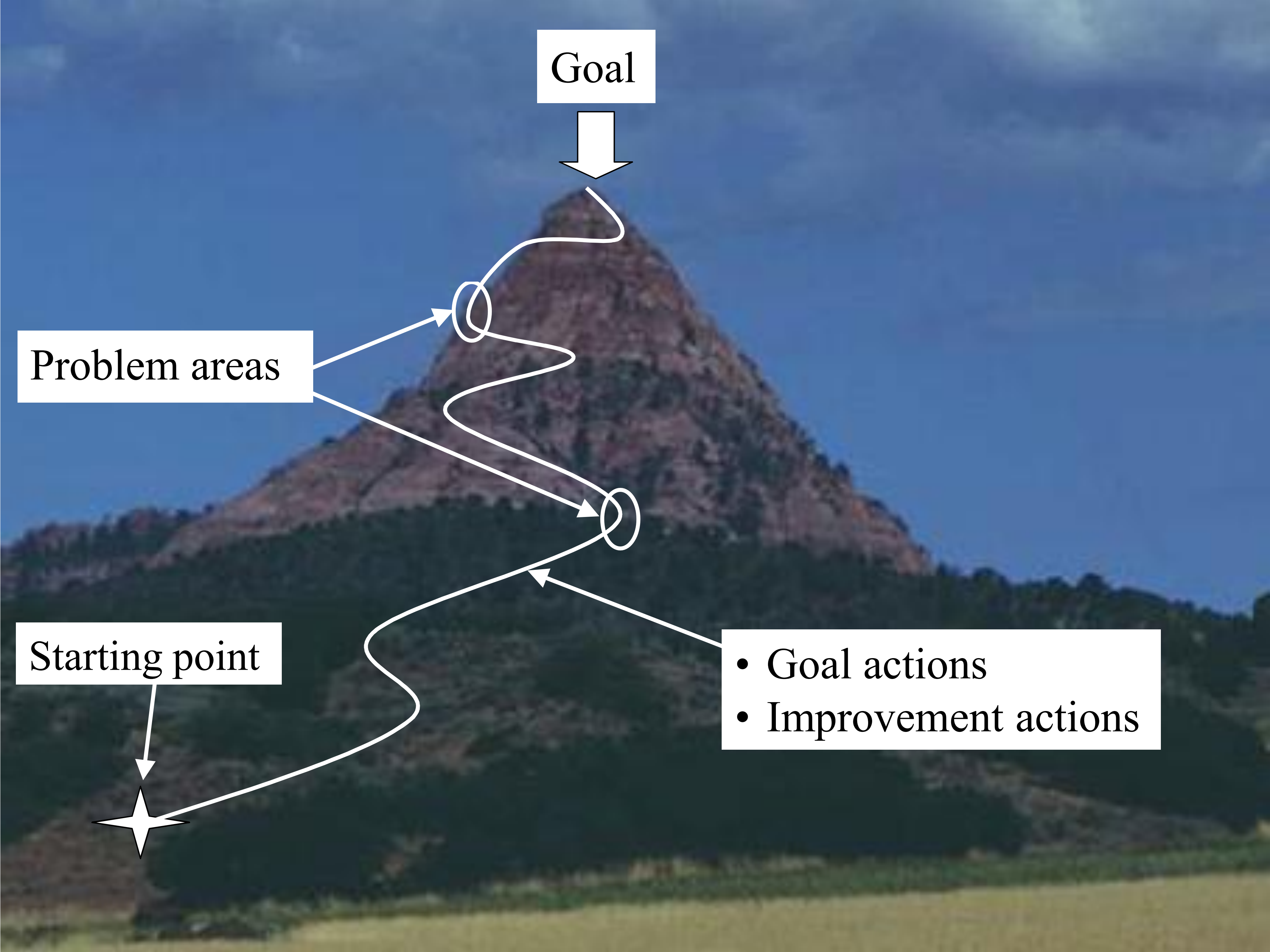
Starting point

A Solution

Goal-problem-centric improvement

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





Goal

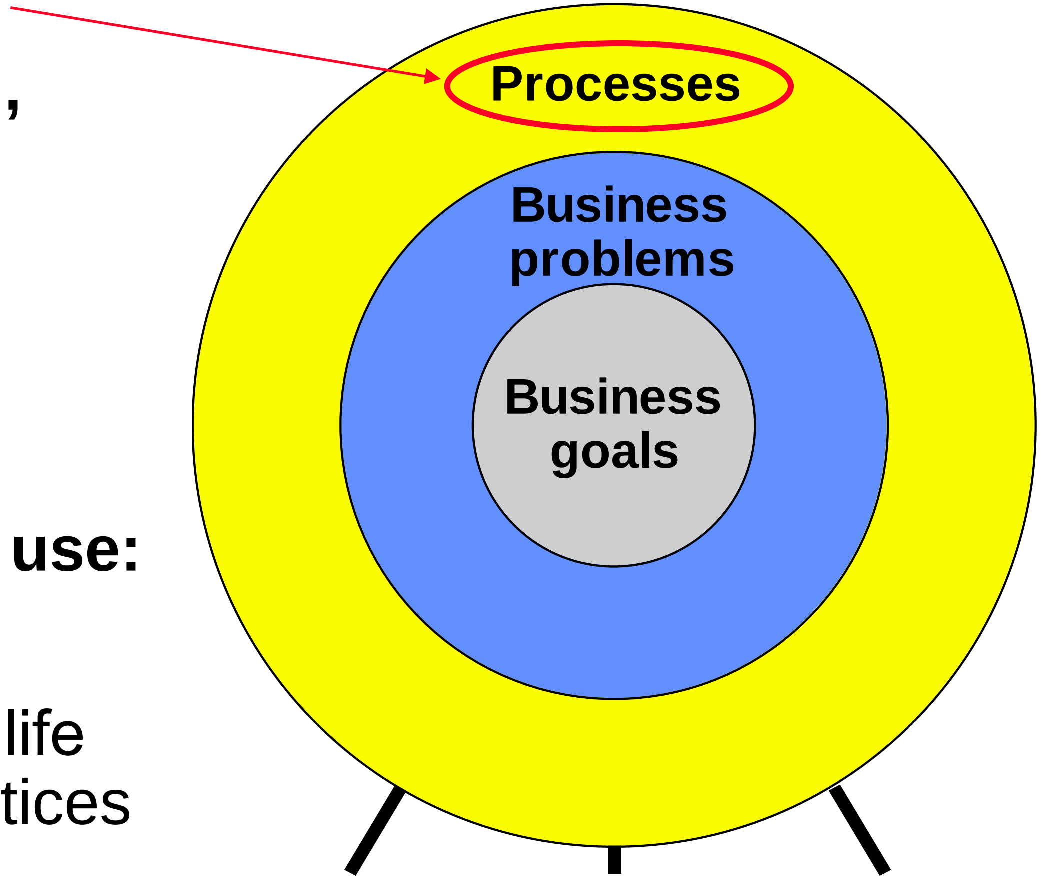
Problem areas

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



Blank slide

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.

Group the Problems Related to Each Goal - 3

Goal	Problem	Problem Description
3. Reduce rework	Problem 3	Quality department does not have training in product and test skills.
	Problem 4	Unclear status of changes.
	Problem 5	Lack of resources and skills allocated to design.
	Problem 9	Defect repairs break essential product features.
	Problem 10	Wrong files (for example, dynamic link libraries) are put on CD. Unsure of the correct ones.

4. Ensure That the Goals and Problems Are Compelling - 1

If there is a big enough reason, people will change in a heart beat!

Pain

PI

Pleasure

- Crisis
- Business loss
- Unhappy customers

- High development cost
- Low profit
- Leadership

- New business
- Increase volume
- Reduce costs

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.

Making Existing Goals More Compelling

- **If you are promoting an idea:**
  - Ask **WHY (<idea>)** to elicit a more compelling reason
  - e.g., WHY (Level 2) may give:
    - **Meet schedules, less rework, more sanity, happier customers**
- **Make the compelling reason the goal, not the process idea**
  - e.g., **goal**: low maintenance **OR** formal inspection?

If the reason is not compelling enough, action will probably not be taken!

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)

5. Set Goal Priorities

Goal	Relative Benefit of Goal (1-10)	Relative Cost of Goal (1-10)	Priority (Benefit/ Cost)	Phase* (1,2,3)
2. Deliver product *x* by mm/dd/yy.	10	4	2.5	1
1. Meet all our cost and schedule commitments.				
5. Achieve customer rating of 9/10 on product evaluation form.				
3. Reduce rework to less than 20 percent of total project effort.				
6. Keep profits at 15 percent (and costs at the same as last year).				
4. Improve the performance of our core software product. (Target to be defined.)				

***Phase is based on goal interdependencies, logical ordering, or timing**

6. Derive Metrics for the Goals

- **The Goal-Question-Metric (GQM) approach from Basili states that you:**
  - Define the principal goals for your activity
  - Construct a comprehensive set of questions that, when answered, helps assess where you are relative to each goal
  - Define and gather the data required to answer these questions

Goal-Question-Metric Approach - 1

Goal	Questions	Metrics
Meet all our cost and schedule commitments.	Are we spending the planned number of hours on the project to complete it? Are we hitting our milestones?	Planned versus actual effort for each project. The number of days each milestone is early or late.
Deliver product X by mm/dd/yy.	Are we spending the planned number of hours on the project to complete it? Are we hitting our milestones?	Planned versus actual effort for each project milestone. The number of days each milestone is early or late.
Reduce rework to less than 20 percent of total project effort.	How much time do we spend on rework now? How does this compare with our development time and are we improving?	Percentage of project time spent on rework.
	How many defects do we have in the product during design and coding?	Defect density: Number of defects found per unit size of work product (e.g., number of pages of design, number of lines of code).

Goal-Question-Metric Approach - 2

Goal	Questions	Metrics
Achieve customer rating of 9/10 on product evaluation form.	How satisfied are they now? Are we improving?	Annual customer satisfaction survey.
Keep profits at 15 percent (and costs at the same as last year).	What is our profit? Is it getting better or worse?	Annual net profit.

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.

Using a Process Appraisal to Obtain a Problem List

- **A scalable data collection method for groups of ~5-150 people, that results in:**
  - a list of strengths and highest-priority problems (& maturity rating)
  - buy-in for the problems
  - buy-in for process improvement direction
- **Surfaces key problems that might not have been visible before:**
  - e.g.,communication, systems engineering, PI implementation
- **Raises awareness of key issues facing the organization**
- **Brings management and engineering together**

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. DonÕt always 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

Actions for Two of the Problems - 2

Problem	What actions are needed to address the problems and achieve the goals?
3. Too many features for the six- to nine-month development cycle	Establish a review process with clients to negotiate features for a six- to nine-month development cycle
	Rate each feature based on value to the customer (1–10 points) and cost to develop (1–10 points)
	Establish an incremental delivery plan to phase in lower priority features

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

Framework Elements for Two of the Problems - 2

Reworded for clarity ↓

Problem	Which elements will help the problems and goals listed?
3. Too many features for the six- to nine-month development cycle	Reconcile the project plan to reflect available and estimated resources. (PP sp3.2)
	Identify and analyze project risks. (PP sp2.2)

PP = Project Planning

Progress on Chosen Framework -1

95% map to Level 2

Example Goals

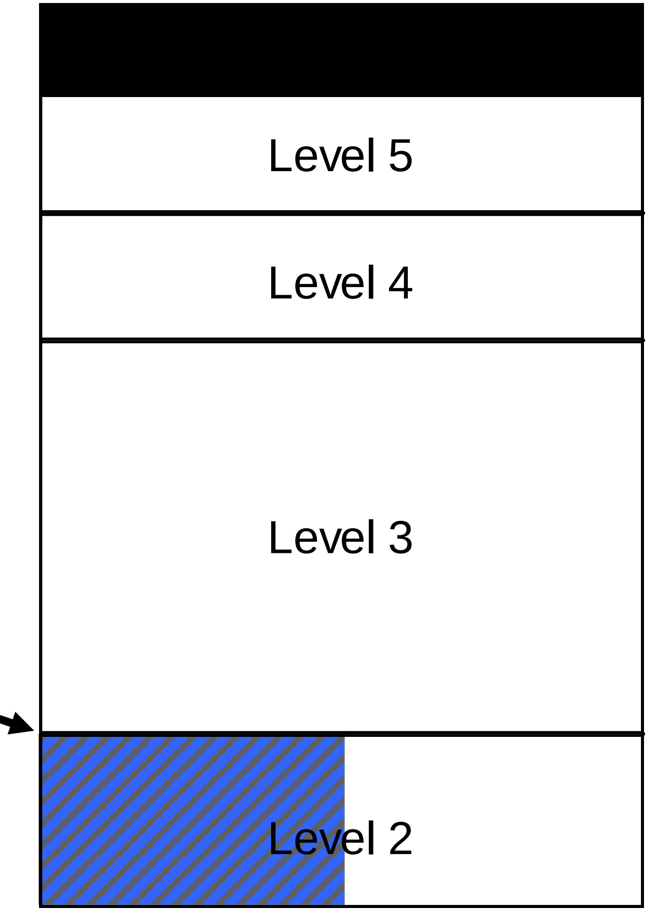
1. Create predictable schedules
2. Successfully deliver product X
3. Reduce rework
4. Improve the performance of our core product
5. Keep customers happy
6. Keep making a profit

Example Problems

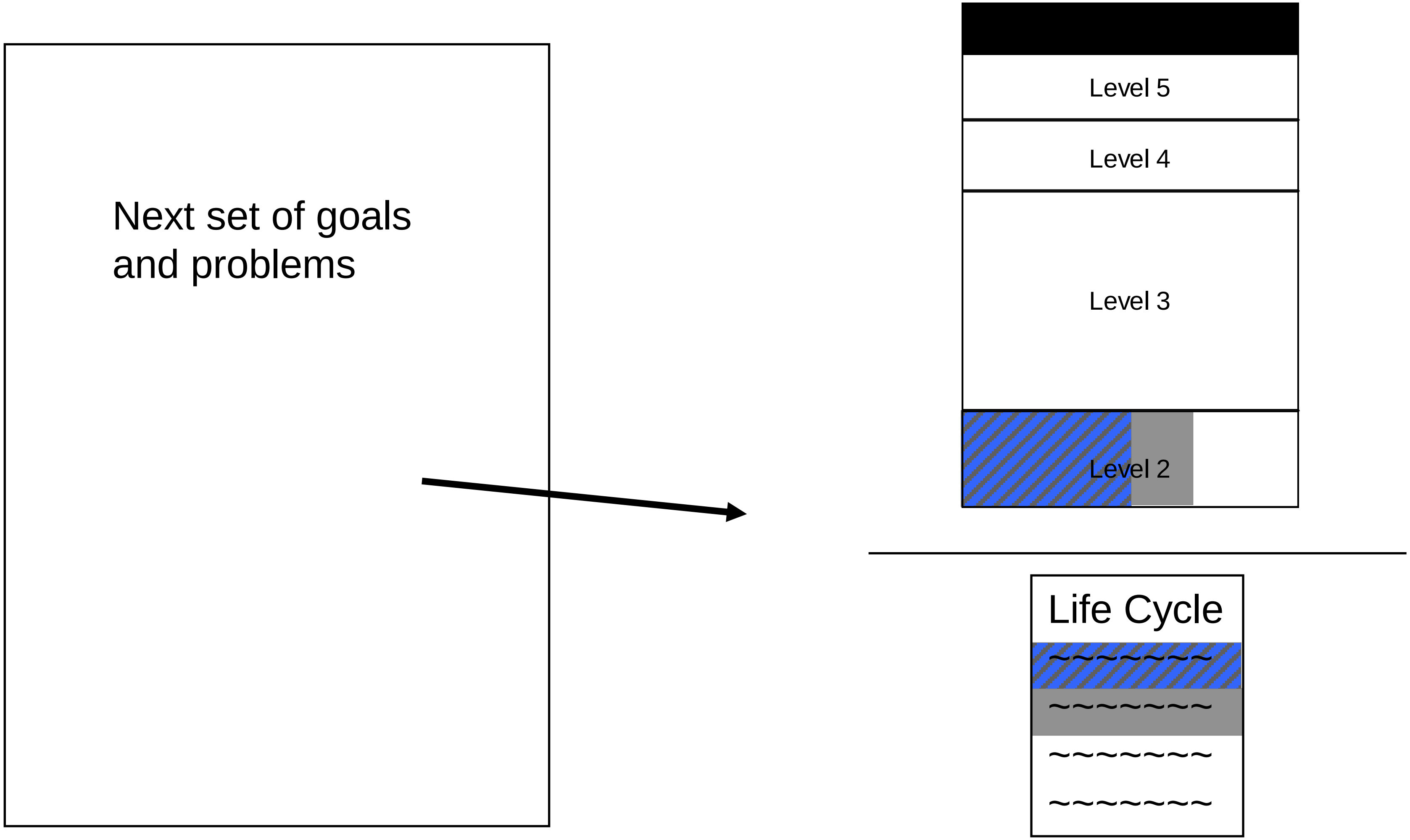
1. Need better requirements. Requirements tracking not in place. Changes to requirements are not tracked; code does not match specification at test tim
2. Management direction unclear for product version 2.3. Goals change often.
3. Quality department does not have training in product and test skills.
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 th correct ones.
11. Revising the project plan is difficult. Items drop off, new things are added, plan is out of date.
12. We donÕ understand our capacity and do not have one list of all the work w 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

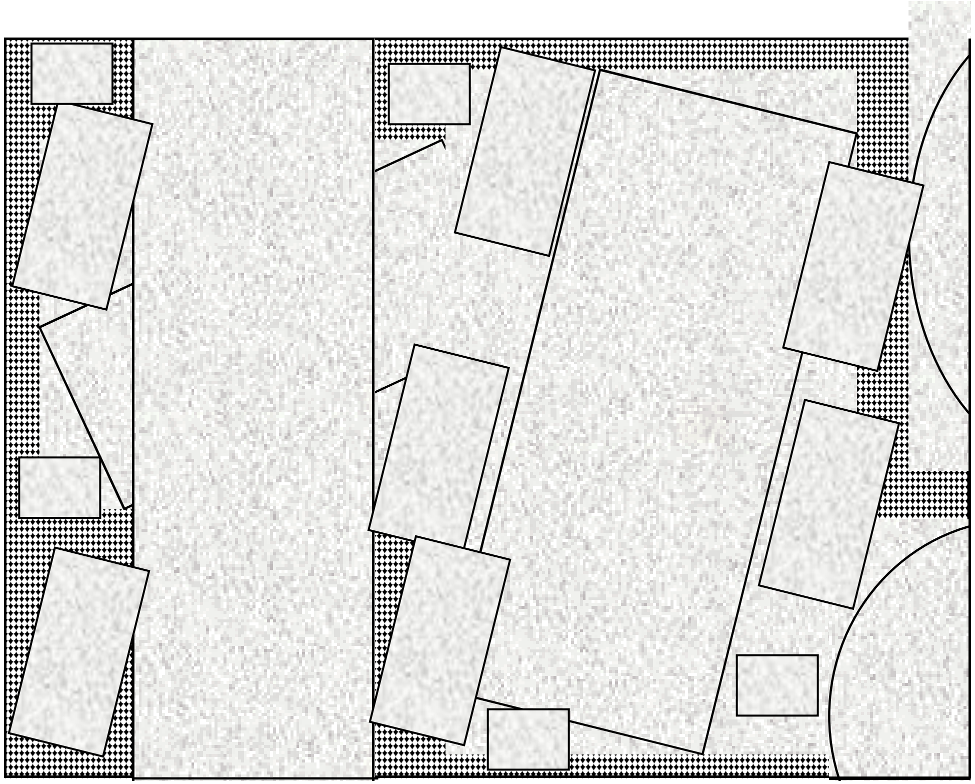
Level 3

Level 2

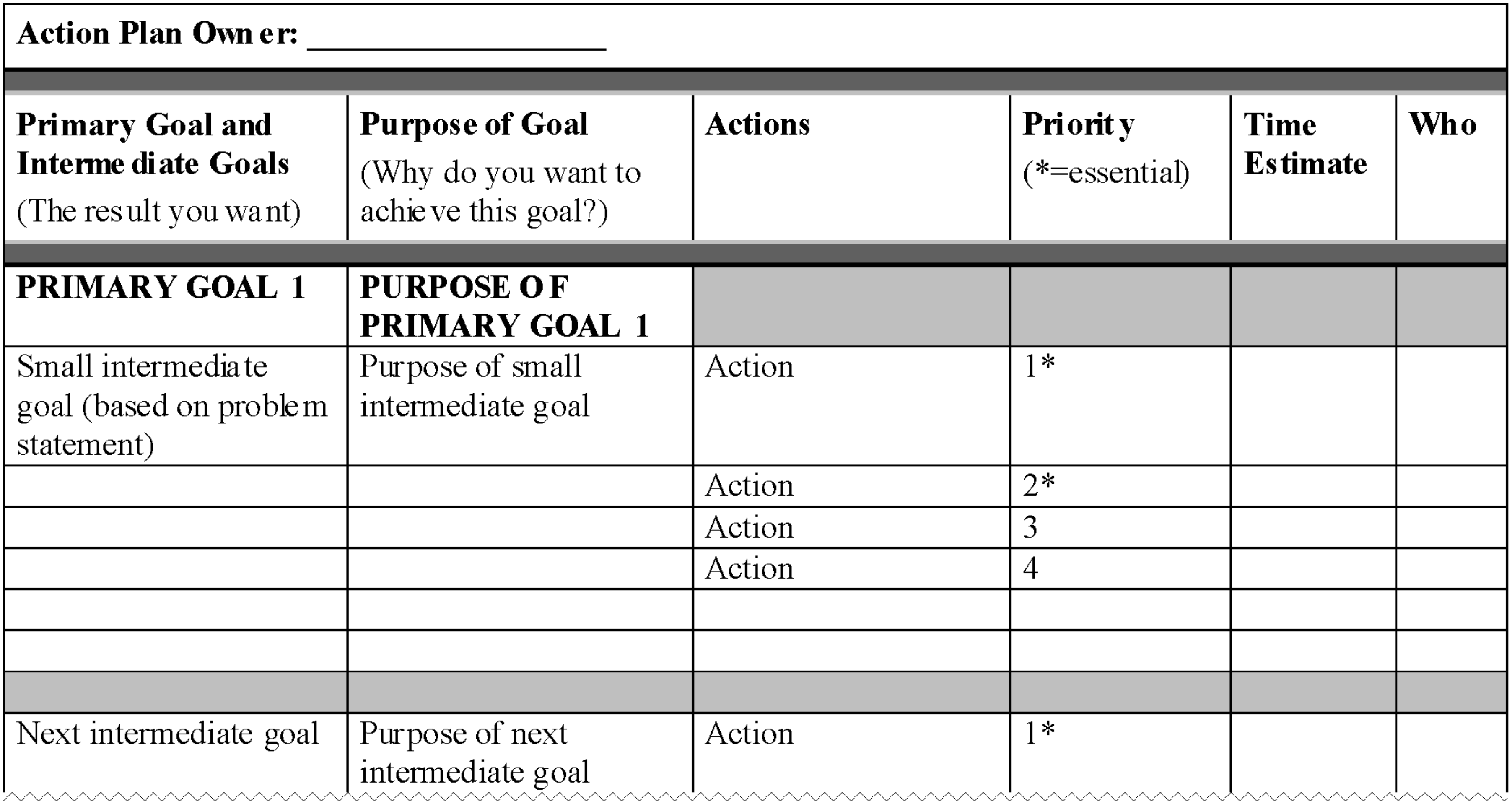
Life Cycle

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

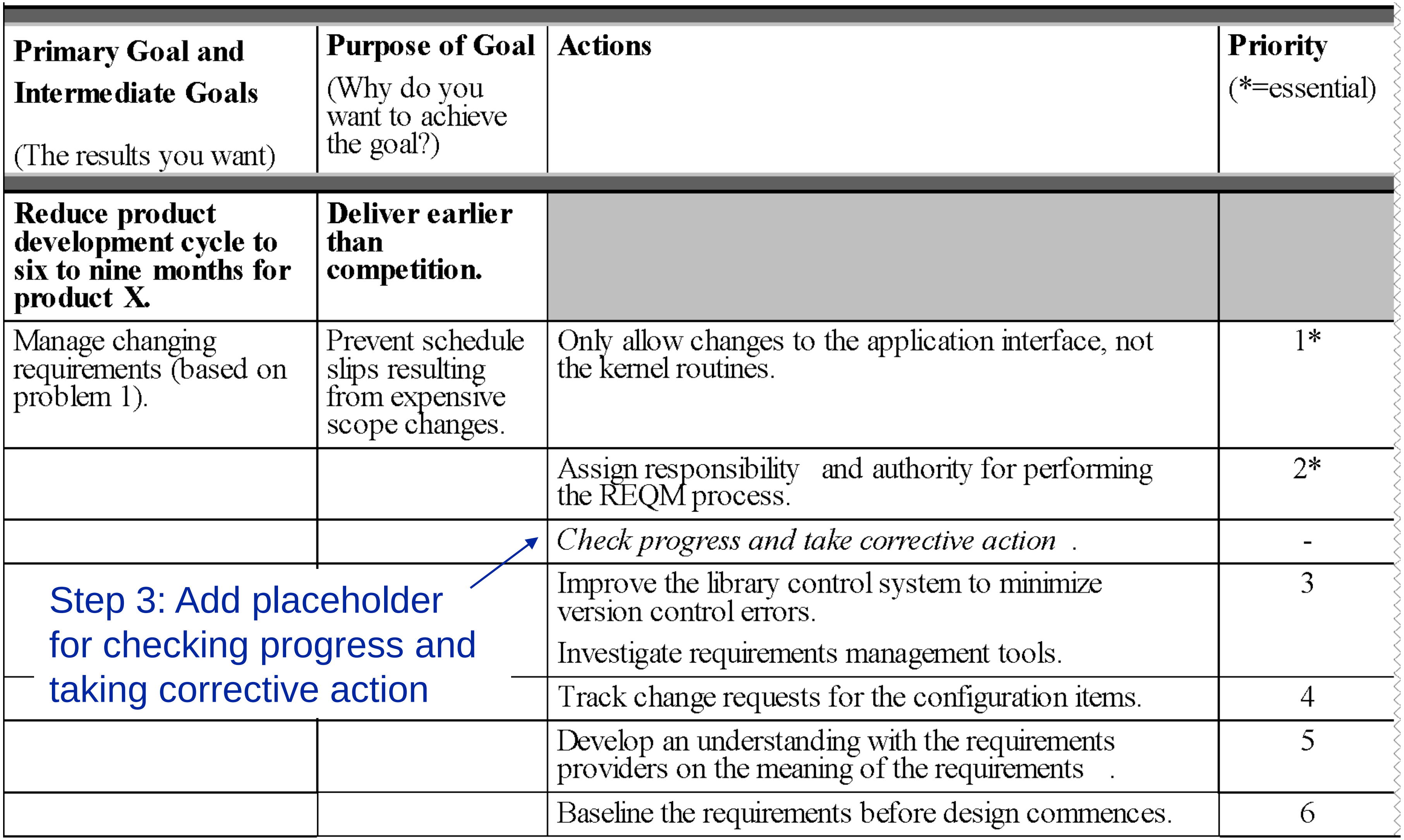


Action Plan Owner: ____________________

Primary Goal and Intermediate Goals (The result you want)	**Purpose of Goal** (Why do you want to achieve this goal?)	**Actions**	**Priority** (*=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.

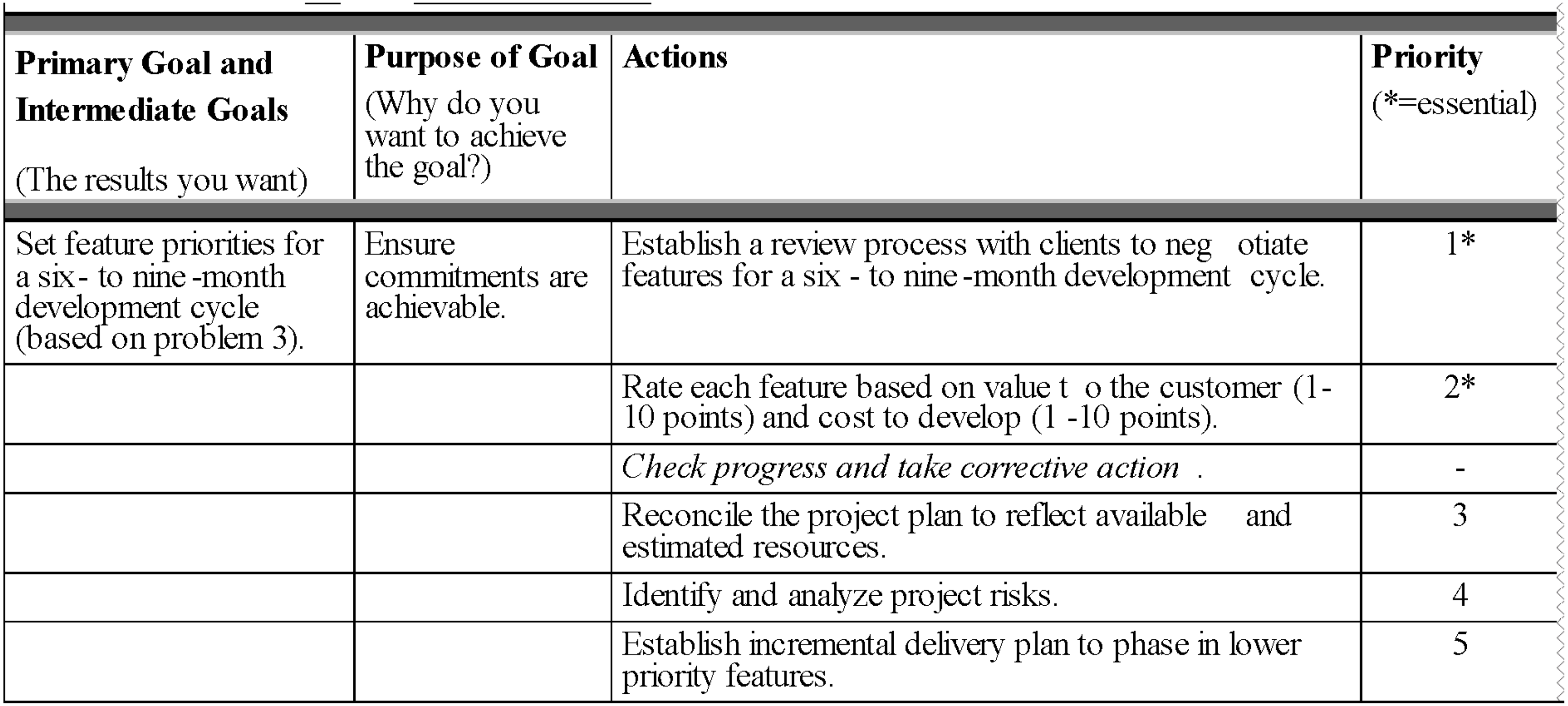
Example Improvement Plan - 1



Primary Goal and Intermediate Goals (The results you want)	**Purpose of Goal** (Why do you want to achieve the goal?)	**Actions**	**Priority** (*=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. 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

Example Improvement Plan - 2



Primary Goal and Intermediate Goals (The results you want)	**Purpose of Goal** (Why do you want to achieve the goal?)	**Actions**	**Priority** (*=essential)
Set feature priorities for a six- to nine-month development cycle (based on problem 3).	Ensure commitments are achievable.	Establish a review process with clients to negotiate features for a six - to nine-month development cycle.	1*
		Rate each feature based on value to the customer (1-10 points) and cost to develop (1 -10 points).	2*
		Check progress and take corrective action .	-
		Reconcile the project plan to reflect available and estimated resources.	3
		Identify and analyze project risks.	4
		Establish incremental delivery plan to phase in lower priority features.	5

Choose Actions That Are Appropriate for the Problem - 1

Problem	Inappropriate and Overly Complex Solution
Unable to get requirements from customers	~~Adopt quality function deployment~~
No time allocated for design	~~Adopt a detailed object-oriented design process~~
Inaccurate estimates	~~Create a new historical database, built from scratch, and available on four platforms~~
Poor-quality products	~~Define a detailed life cycle, containing numerous engineering methods~~

Choose Actions That Are Appropriate for the Problem - 2

Problem	Simpler Solution
Inaccurate estimates	Learn an estimation process that addresses some of the root causes of the inaccurate estimates (for example, the Wideband Delphi method)
	Start collecting actual data for current projects so that they can compare their estimates with actual effort expended

Choose Actions That Are Appropriate for the Problem - 3

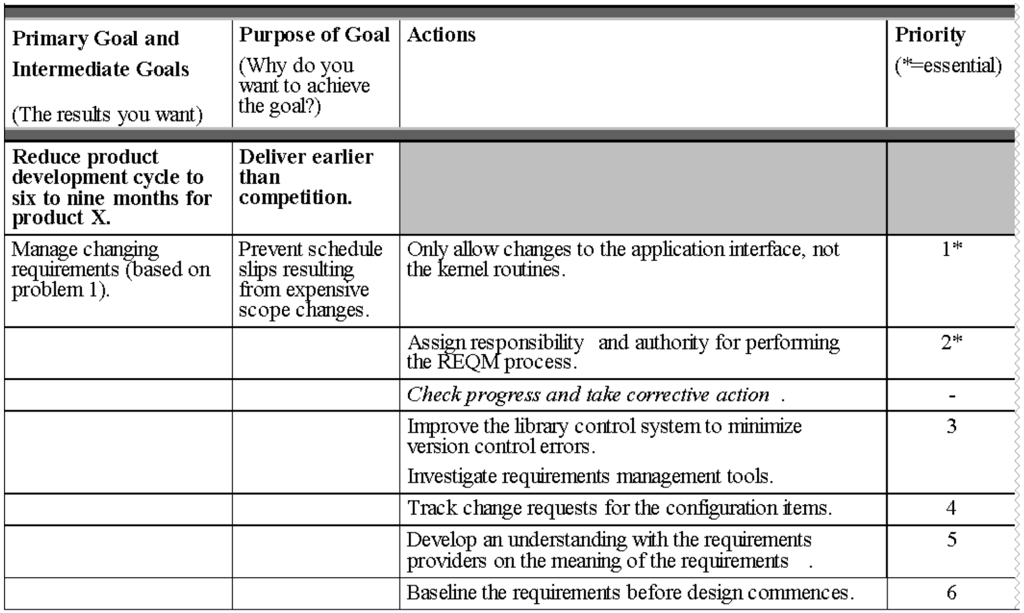
Problem	Simpler Solution
Poor-quality products	Inspect (peer review) all critical documents and code.
	Improve estimation of test time needed.
	Train test engineers in test skills.
	Send test engineers to a customer site to understand how the customer uses the product. Factor this knowledge into the test strategy.

Exercise: Develop an Action Plan

1. **Form project teams**
2. **Select 2-3 goals (and related problems) to develop actions for**
3. **Develop actions:**
   - Brainstorming
   - Select elements from an improvement framework
   - Establish priorities and essential actions

4. **Discuss lessons learned**

Result:



Primary Goal and Intermediate Goals (The results you want)	**Purpose of Goal** (Why do you want to achieve the goal?)	**Actions**	**Priority** (*=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. 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

Developing a Plan

- Scope the Improvement
- Develop an Action Plan
- **Determine Risks and Plan to Mitigate**
  1. Determine Scope of Risk Session
  2. Select the Team and Moderator
  3. Identify Risks
  4. Analyze Risks
  5. Plan to Mitigate
  6. Plan for Periodic Risk Review

What is a Risk?

A risk is anything negative that could happen

1 + 2. Scope, Team and Moderator

- **Scope**:
  - The complete list of goals and problems, or subset
- **Team**:
  - Improvement team
  - People who have done similar improvement projects and tasks
  - Technical experts
  - Customer champions
- **Moderator**:
  - Any team member

3. Identify Risks

Create a list of potential future problems

- **Weak areas such as unknown technology**
  - e.g., tools, vendors, and methods that are new to the team
- **Aspects that are critical to the improvement project**
  - e.g., the timely delivery of a vendor’s training program, continued buy-in from management, and the creation of training materials
- **Problems that have plagued past projects**
  - e.g., loss of essential staff, resistance to change, and changing priorities

4. Analyze Risks

For each risk item:

- **Does the team understand this risk item?**
  - » If necessary, split into separate risk items

    "Lack of management buy-in" is replaced by:

    -> "Jane could decide that the new method is not beneficial"

    -> "Robert (VP) does not see any gains from money spent"
- **Discuss and determine its scope:**
  - » What would the consequences be if this risk item did occur?
- **Determine what the impact (I)** would be if the worst happened, using a scale of **one to ten.**
- **Determine how likely (L)** it is that the risk item will occur, using a scale of **one to ten.**
- **Determine the priority (P)** of the risk items using **impact x likelihood.**

Risk Analysis Example

Risk Items	Consequence	L	I	P = L x I
Management buy-in for improvement diminishes	Improvement program fails	9	10	90
Management changes priorities before we complete any milestone	Improvement program loses credibility	9	9	81
New requirements management tool has long learning curve	Developers give up in frustration	9	8	72
Library control person might leave	Wasted time training a new person	7	8	56
New group to manage baseline changes is not accepted by project managers	Duplication of effort or baseline changes are not managed	6	9	54
Creation of specialized training materials for new staff takes too long	Improvement implementation delayed	5	7	35
Requirements management tool is delivered to us late	Pass up the opportunity to try the tool	4	3	12

5. Plan to Mitigate

- **Select the most important risk issues, such as the top 2-3, or top 20%**
- **Brainstorm on actions that could be taken to reduce the likelihood of the risk item occurring** (risk mitigation)
  - Make actions specific
- **Brainstorm on actions that could be taken to reduce the impact if the risk item does occur** (risk contingency planning)
- **Decide which actions to pursue**
- **Select a person to be responsible for each action chosen**
- **Document the information in the risk management plan**

Risk Planning Example

Risk item:

- Management buy-in for improvement diminishes

Possible actions to reduce likelihood of risk item:

- Ensure that the improvement program addresses the management team’s problems and goals
- Establish a steering committee to oversee the improvement effort
- Publicize early results to management
- Provide four funding options for the improvement program: full-time, part-time, short bursts, and investment spread over two years

Possible actions to reduce impact of risk item if it does occur:

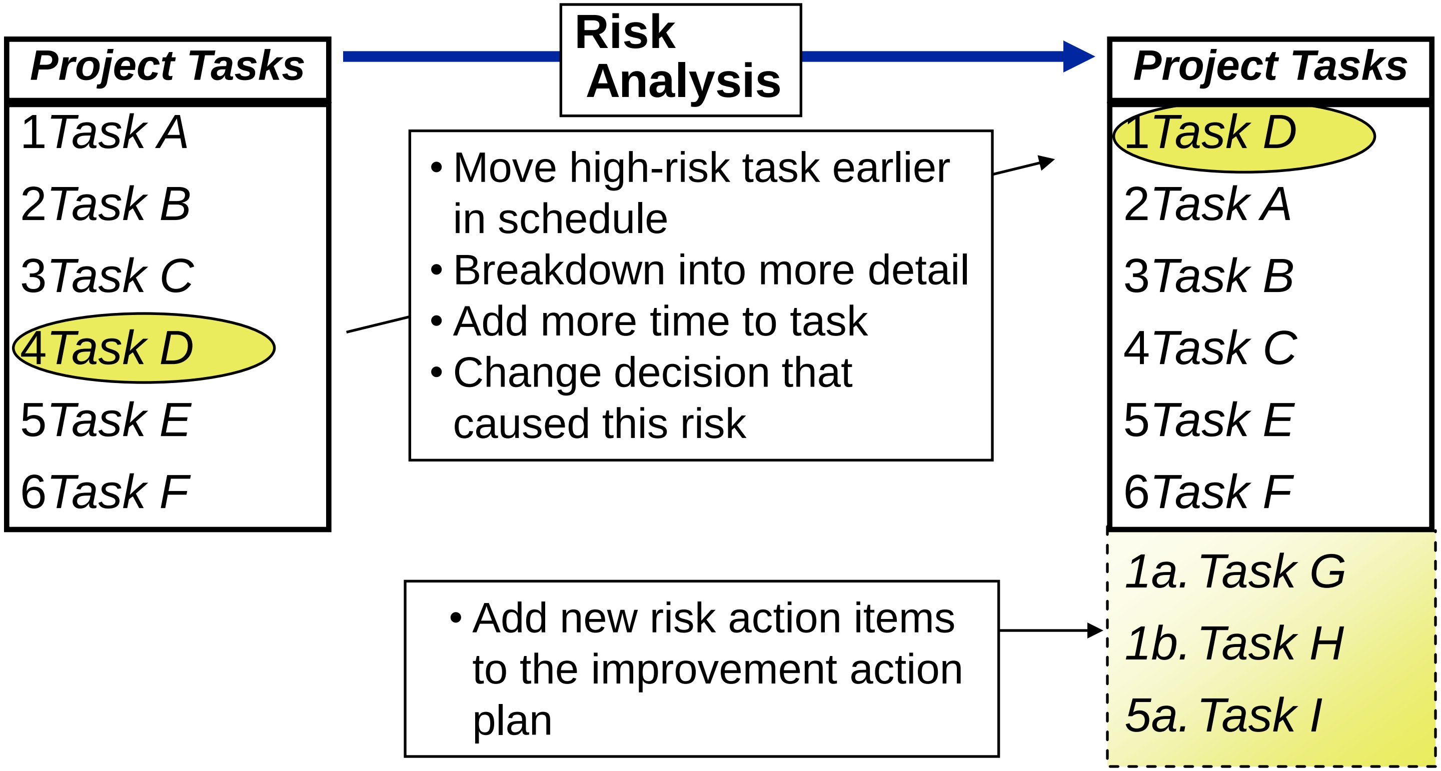
- Determine improvements that can be made at a project level without major funding
- Explain the problems and goals that will not be addressed because of reduced funding

Risk Management Plan

Risk Item	Consequence (If Risk Item Does Occu r)	L	I	P	Actions to Reduce Likelihood of Risk Occurring	Actions to Reduce Impact if Risk Does Occur	Who is Responsible for These Actions?	When Actions Should be Complete	Status of Action

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

Using the Risk Information



Project Tasks

1 *Task A*

2 *Task B*

3 *Task C*

4 *Task D*

5 *Task E*

6 *Task F*

Risk Analysis

- Move high-risk task earlier in schedule
- Breakdown into more detail
- Add more time to task
- Change decision that caused this risk

- Add new risk action items to the improvement action plan

Project Tasks

1 *Task D*

2 *Task A*

3 *Task B*

4 *Task C*

5 *Task E*

6 *Task F*

1a. Task G

1b. Task H

5a. Task I

The improvement team should direct its attention to the higher risk, rather than the easier (lower risk) tasks

6. Plan for Periodic Risk Review

Purpose:

To determine if the identified risks have changed and update the plan accordingly

- Be sure that periodic risk reviews are held to monitor the risks identified
- Establish how often risks should be reviewed (once a month is typical)
- Risk reviews can be incorporated into existing project status and phase reviews
- Update the list based on risk review sessions

Exercise: Determine Risks & Plan to Mitigate

1. **Form project teams.**
2. **Determine the scope of the risk session and establish the goal (i.e., which goals and problems).**
3. **Determine the risks and priorities. Consider your assumptions and improvement actions.**
4. **Determine the action items for top 2-3 risks.**
5. **Discuss lessons learned.**

Result:

Risk Items	Consequence	L	I	P = L x I
Management buy-in for improvement diminishes	Improvement program fails	9	10	90
Management changes priorities before we complete any milestone	Improvement program loses credibility	9	9	81
New requirements management tool has long learning curve	Developers give up in frustration	9	8	72
Library control person might leave	Wasted time training a new person	7	8	56
New group to manage baseline changes is not accepted by project managers	Duplication of effort or baseline changes are not managed	6	9	54
Creation of specialized training materials for new staff takes too long	Improvement implementation delayed	5	7	35
Requirements management tool is delivered to us late	Pass up the opportunity to try the tool	4	3	12

Action Items

~~~~~~
~~~~~~

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**

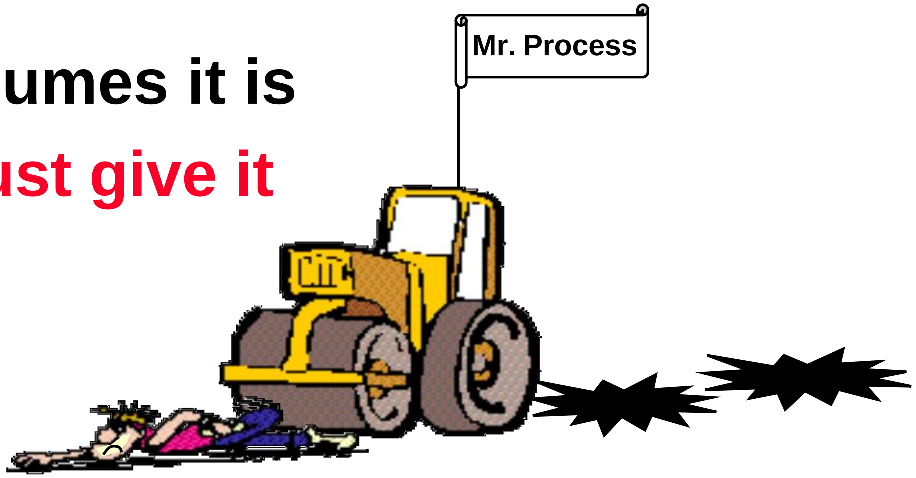
Implementing the Plan

“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



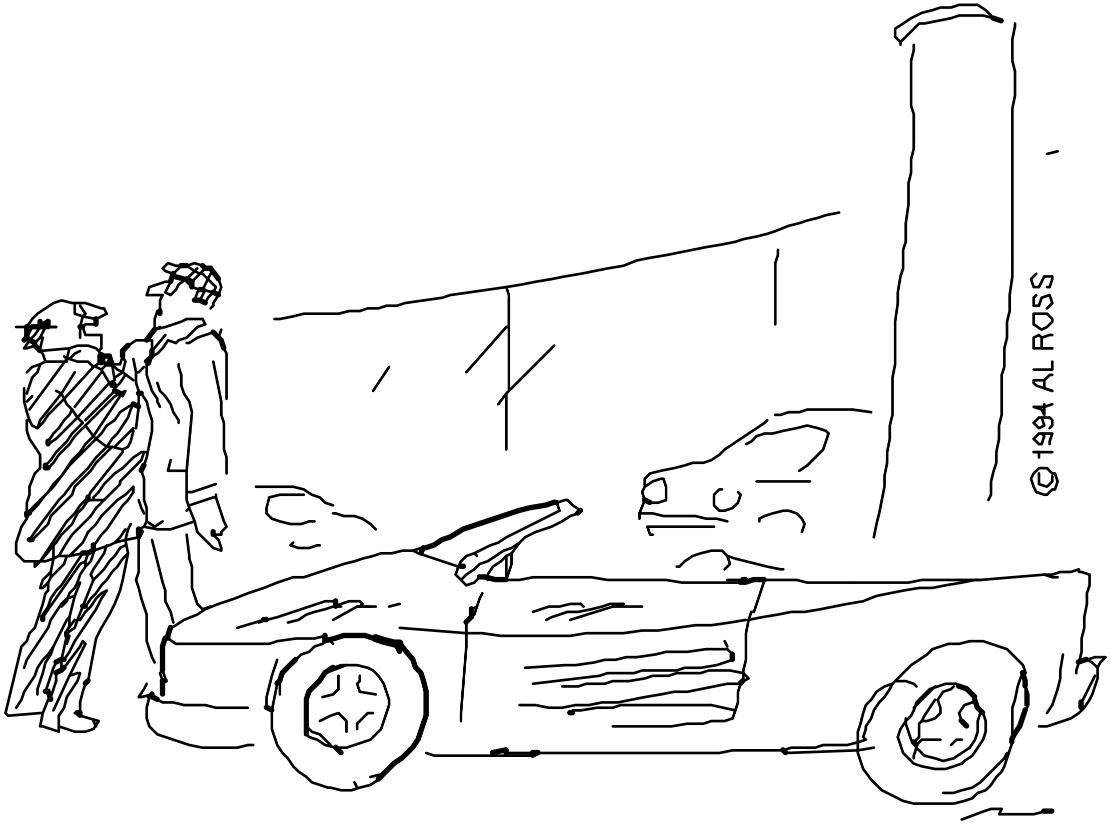
Implementing the Plan

- – Sell Solutions Based on Needs
- – Work with the Willing and Needy First
- – Keep Focused on the Goals and Problems
- – Align the Behaviors of Managers and Practitioners
- – Avoiding a Documentation Glut

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*

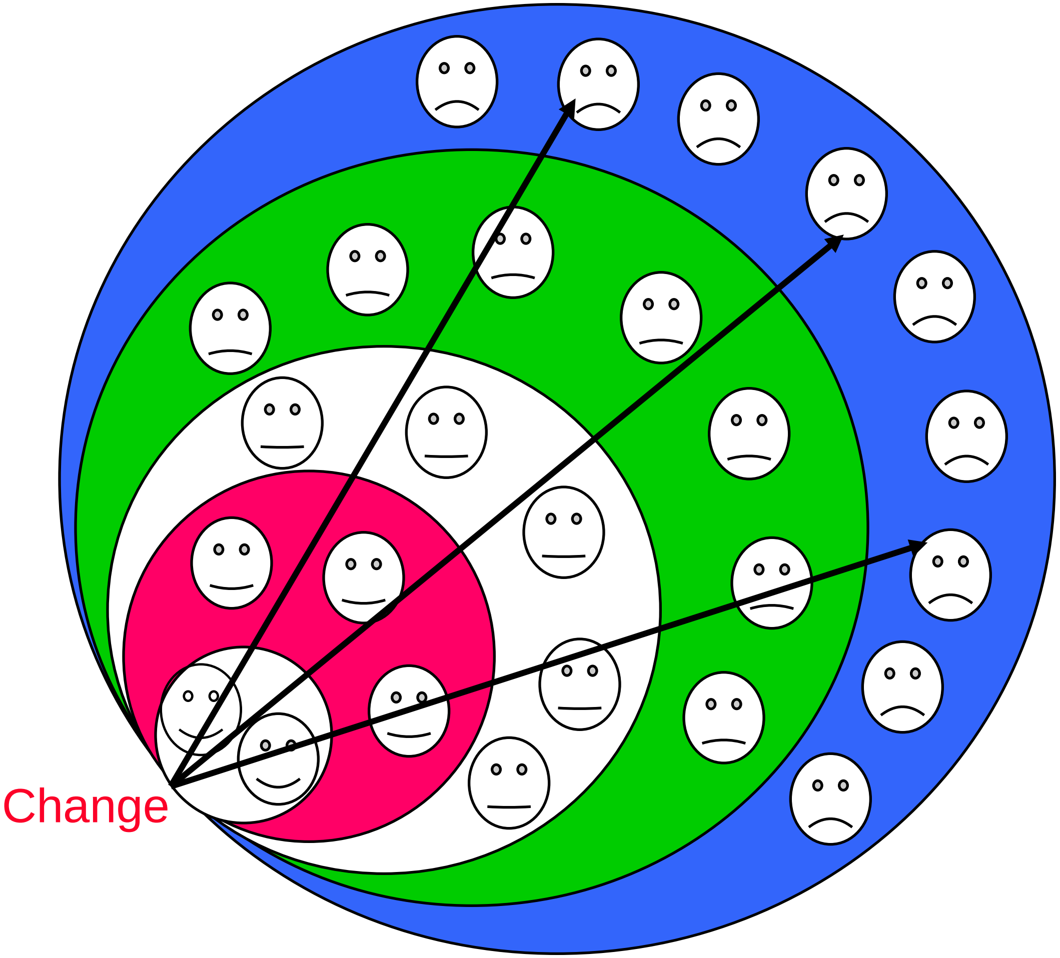
- **Forget** what you are selling
- **Understand** what the customer wants in his/her terms
  - Problems and goals
- Determine the **match** with what you have and what the customer wants
- **Solve** the customer’s problem
  - may be a standard or customized solution

Implementing the Plan

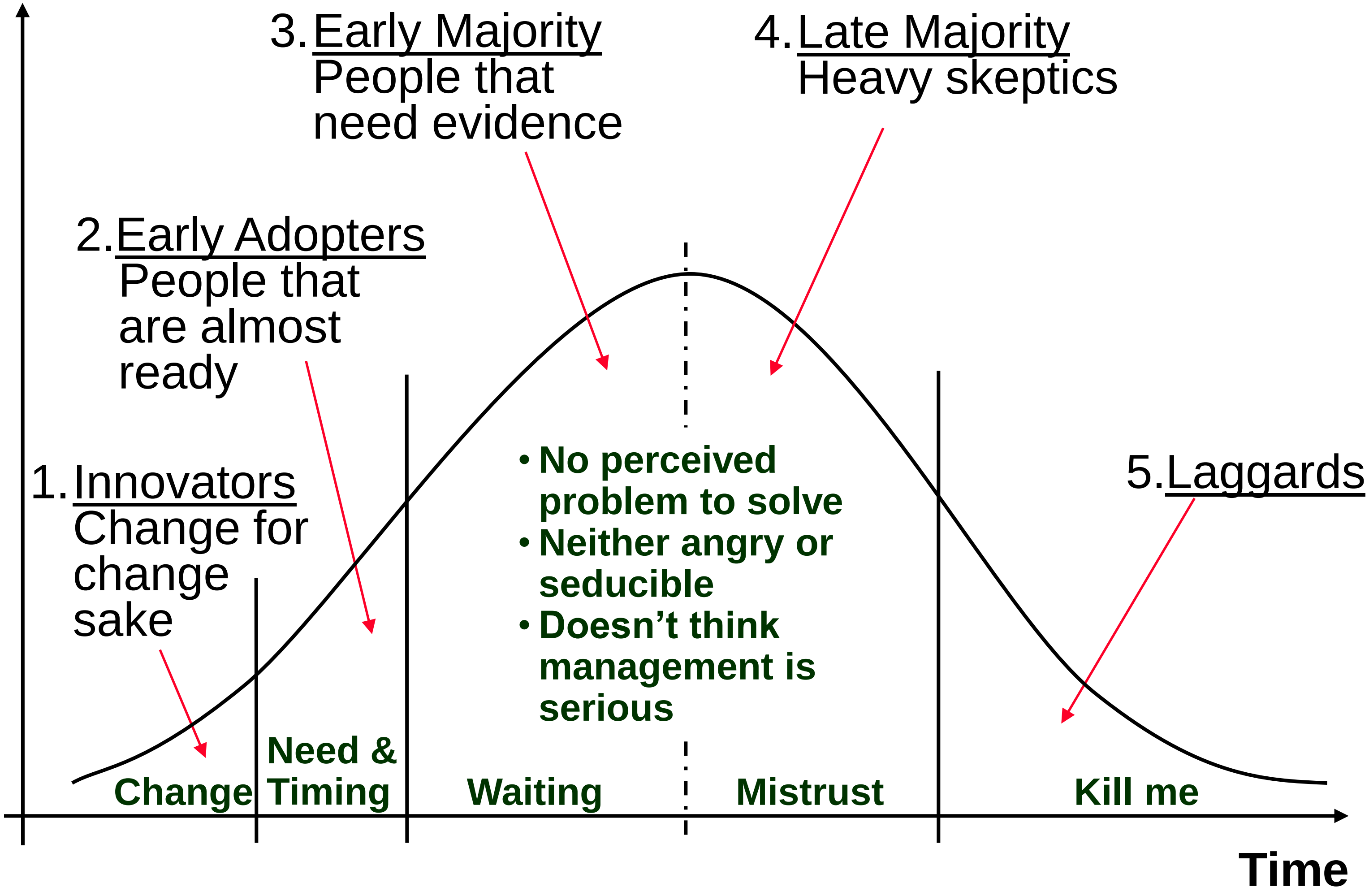
- – Sell Solutions Based on Needs
- – Work with the Willing and Needy First
- – Keep Focused on the Goals and Problems
- – Align the Behaviors of Managers and Practitioners
- – Avoiding a Documentation Glut

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?

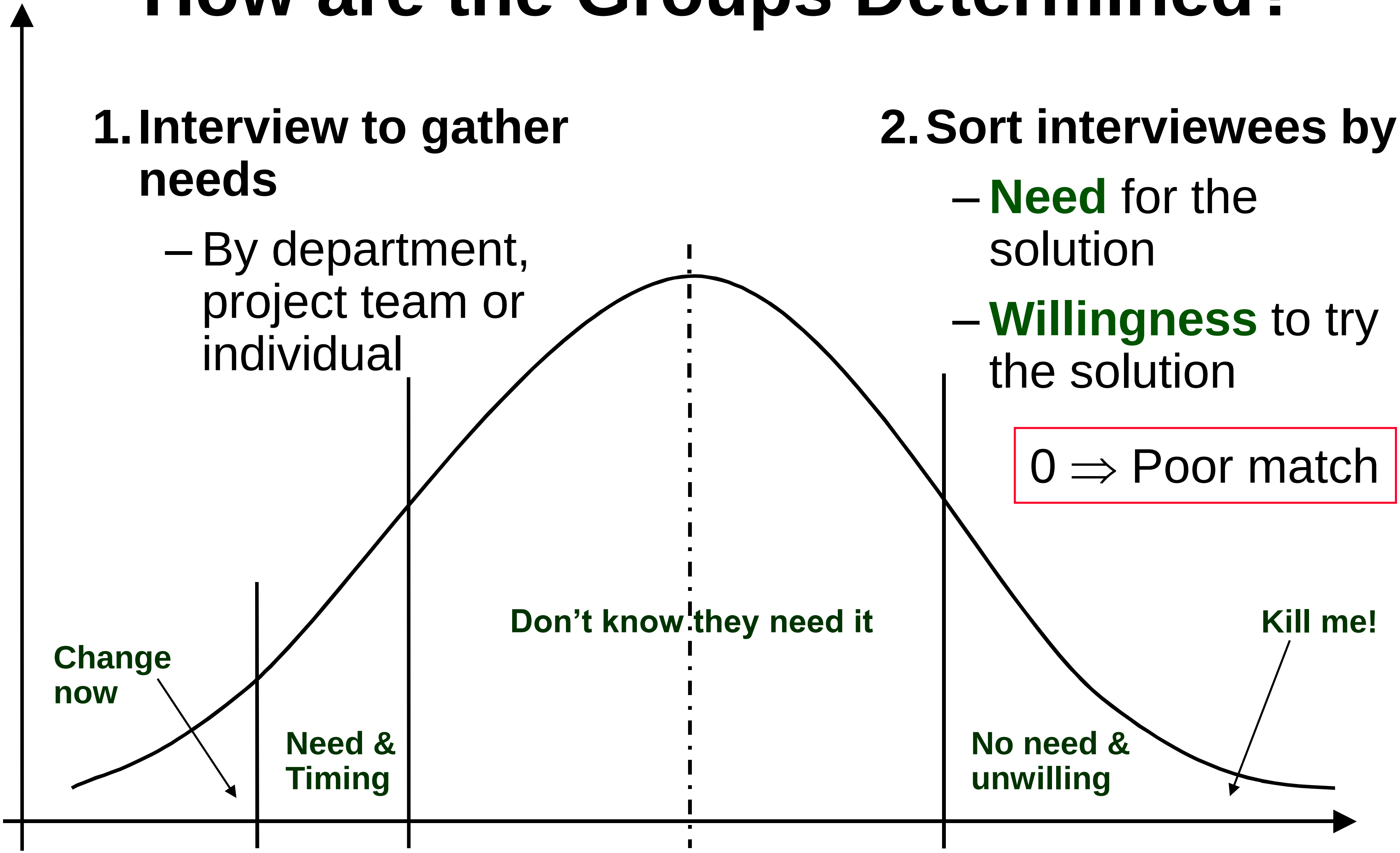

3. Early Majority
People that
need evidence
4. Late Majority
Heavy skeptics
2. Early Adopters
People that
are almost
ready
1. Innovators
Change for
change
sake
5. Laggards
• No perceived problem to solve
• Neither angry or seducible
• Doesn't think management is serious
Change
Need &
Timing
Waiting
Mistrust
Kill me
Time

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



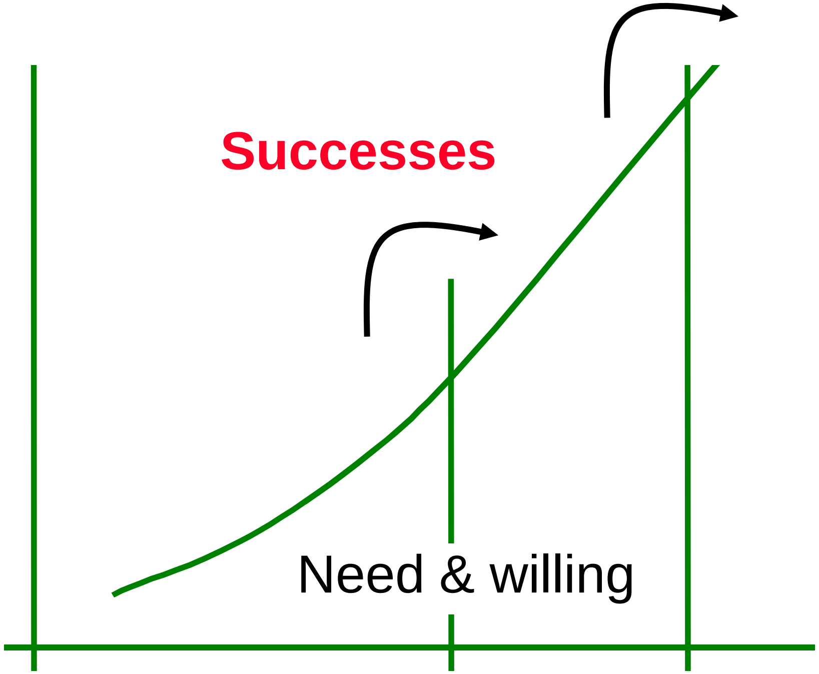
Three Uses of the Adoption Curve

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

Use 1: Increase Speed of Deployment

Speed comes from:

- **Increasing motivation to adopt - based on need**
- **Decreasing resistance - based on willingness**
- **Using previous successes to influence the next group**



Successes: Getting the Word Out

- How to reach your audience

Spoken word

brown bags, seminars, department meetings, discussion groups

Written promotional material

brochures, articles, reports, newsletters

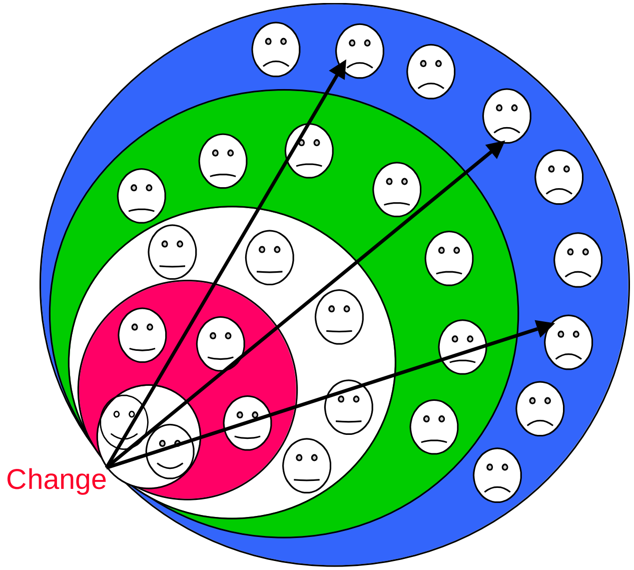
Individual contact

consulting, facilitation

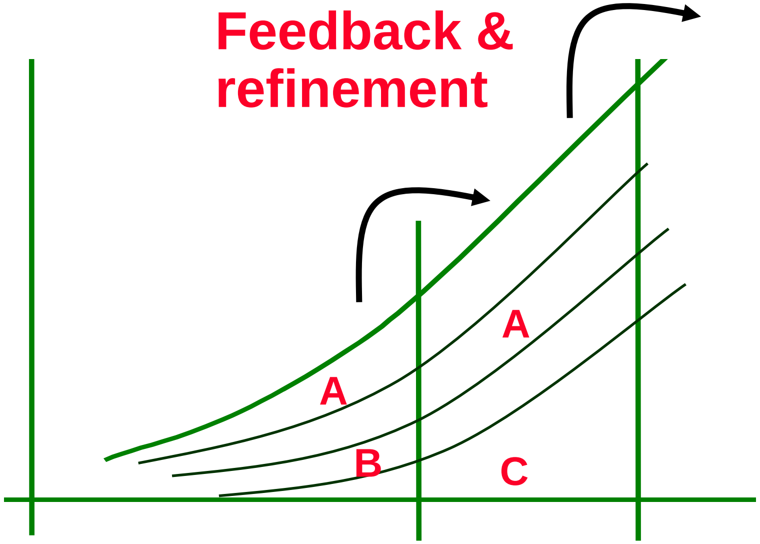
Written processes

job aids, guidelines

Use 2: Reduce the Risk of Failure



Innovators and early adopters can provide specific requirements for and feedback on early versions of the solution

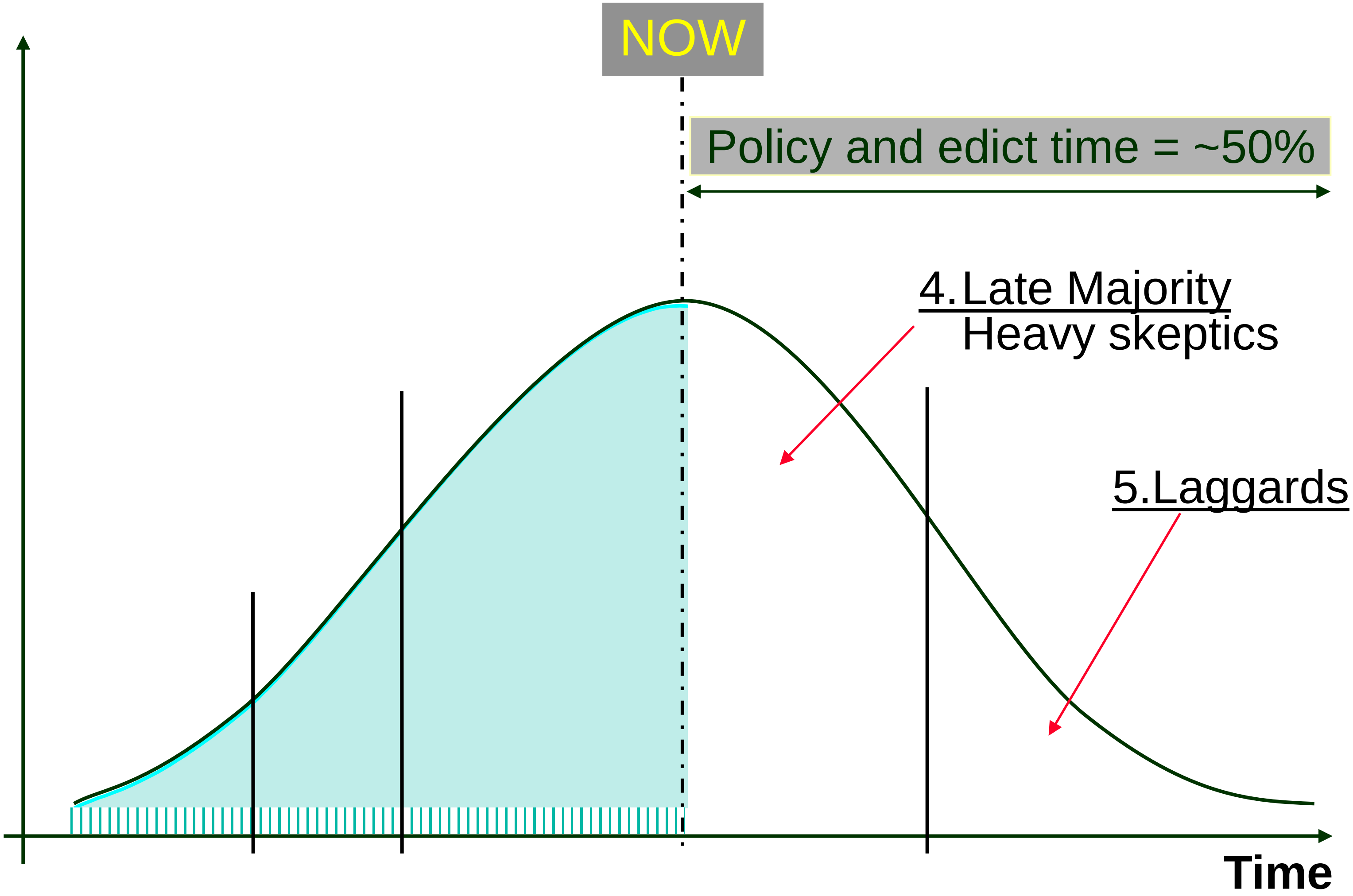


For example:

Planning solution for 3 teams

- Team A: Estimation
- Team A: Negotiation
- Team B: Risk management
- Team C: Metrics / tracking

Use 3: Developing a Policy & Issuing an Edict - 1



Use 3: Developing a Policy & Issuing an Edict - 2

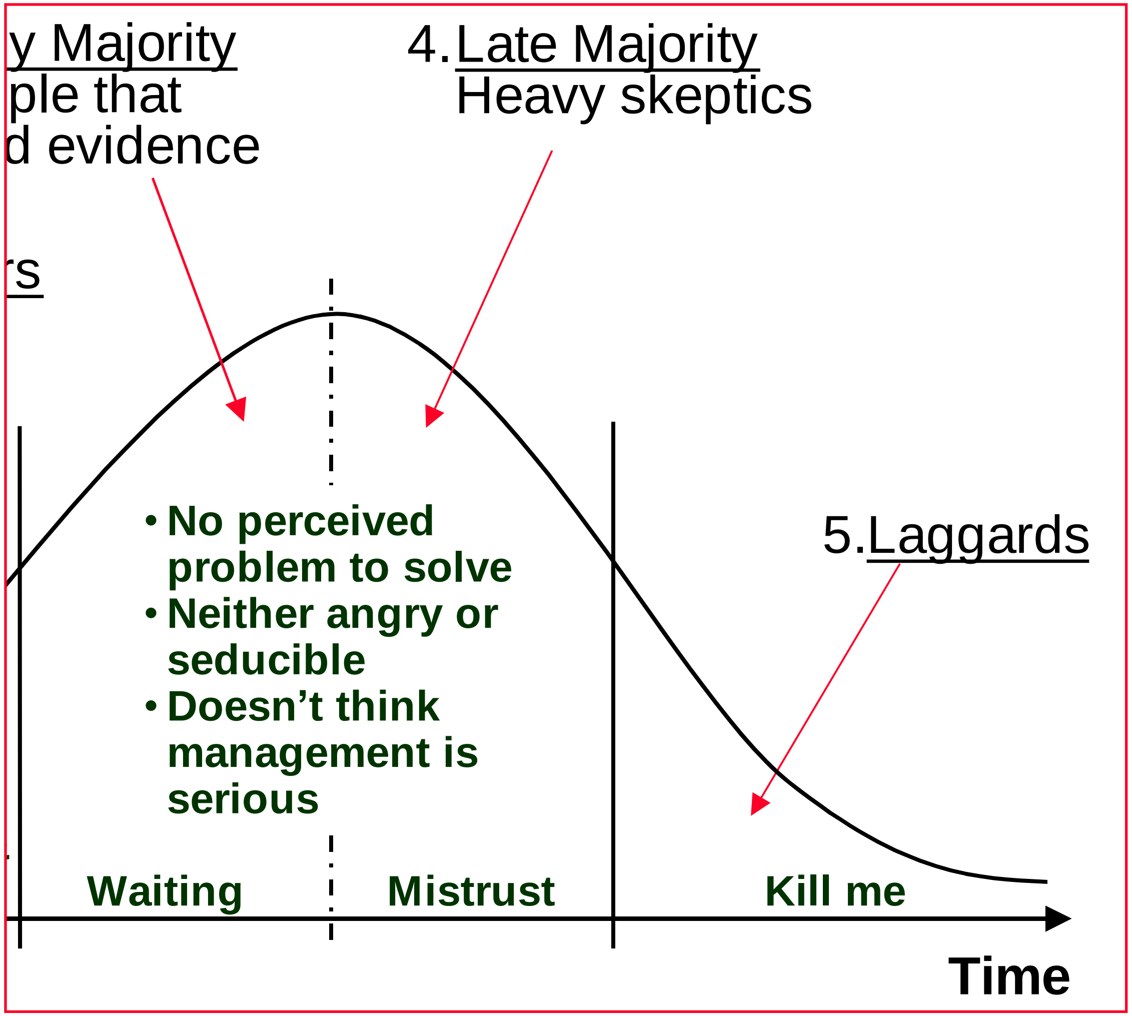
- **Policy states:**
  - When and where a practice should be used.
- **In the beginning:**
  - You might not have any idea!

Wait until you get some experience and <u>feedback</u>

- **Edict states**:
  - "Do it now, this is important."
- **In the beginning**:
  - You don't necessarily have proof or credibility.

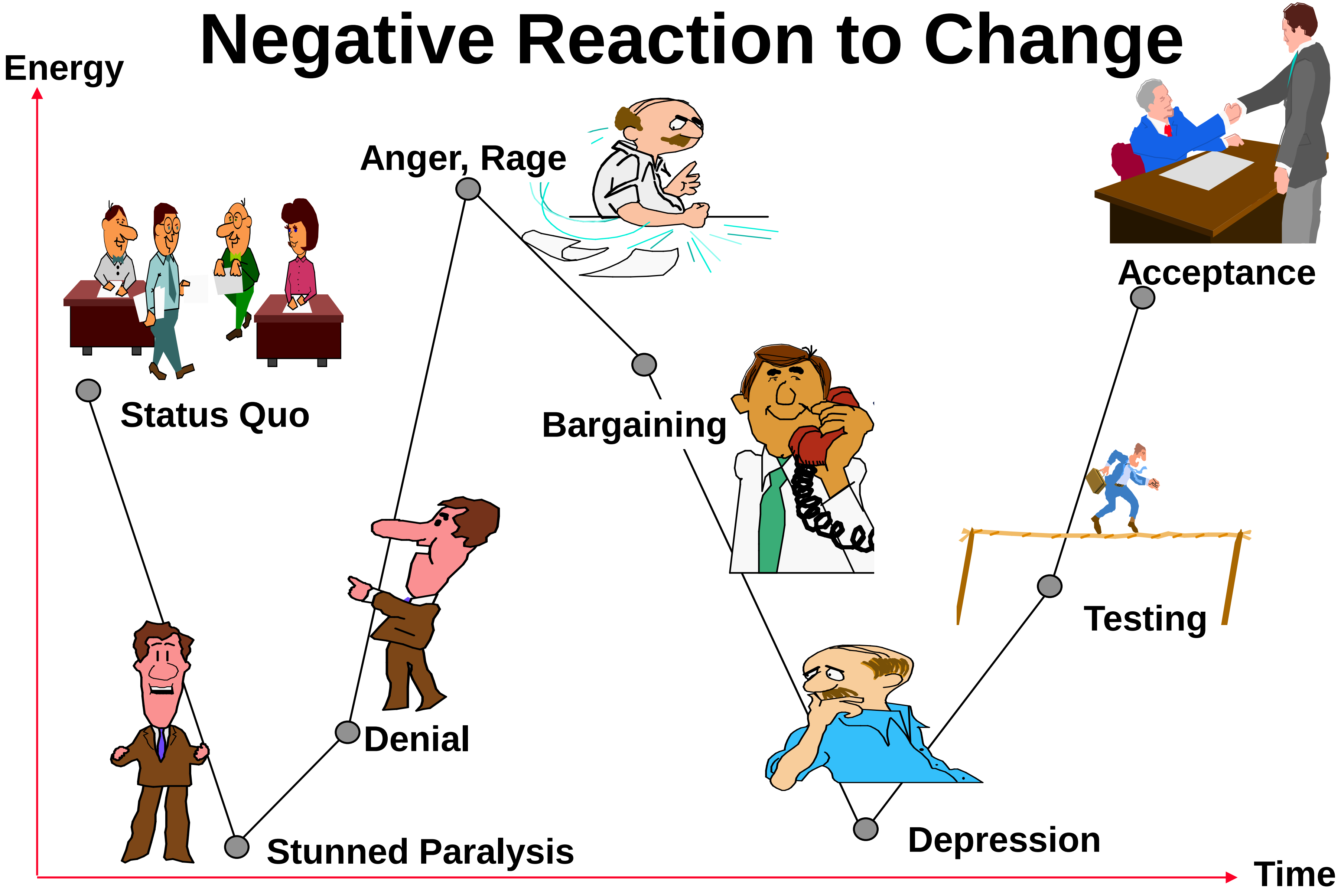
Wait until you get some experience and <u>ownership</u>

Resistance



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

Negative Reaction to Change



Energy

Status Quo

Stunned Paralysis

Denial

Anger, Rage

Bargaining

Depression

Testing

Acceptance

Time

Overcoming Resistance

Resistance has two common causes:

1. **It is not apparent to the person resisting that your solution will meet his or her current needs**
2. **The person believes that your solution brings more pain than benefit; examples of pain include:**
   - embarrassment (if the change is unsuccessful)
   - wasted time using a poorly constructed solution
   - fear of stepping into the unknown (when the status quo is comfortable)

Overcoming Resistance - Needs -1

- Address the first common cause by:
  - identifying and clarifying the needs of your audience
    - » What is the problem and what are they trying to accomplish?
    - » Do they understand your proposed solution and is this an appropriate time to adopt the idea?
    - » What are their concerns regarding costs (or effort / timing)?
- If your solution does not match the need, then say so, and investigate other solutions that do
- If the issue is timing or cost:
  - determine a more appropriate occasion to deploy the new skill, or
  - propose a smaller, more economical solution

Overcoming Resistance - Needs - 2

Understand the benefit of the current behavior. The new behavior must give them at least the benefit of the old.

Example:

- **Quarterly product releases:**
  - Keeps customers happy
  - Helps keep team focused
  - Generates constant revenue
- **Semi-annual releases may also:**
  - Keeps customers happy
  - Helps keep team focused
  - Generates constant revenue

Adoption Requires Leverage

(Some identified pain, or missing pleasure, associated with the current behavior)

"I'll do this peer review stuff if it can help me make tomorrow's demonstration for the CEO"

Unless you find that area of leverage (pain or pleasure), the change may never happen

Overcoming Resistance - Beliefs - 1

Events lead to beliefs (things we feel certain about). Beliefs combined with values (what is important) lead to behaviors.

Event or Information
Bad code review

+

Beliefs
Reviews grade the author

Values
•Deadline
•Ego
•Respect

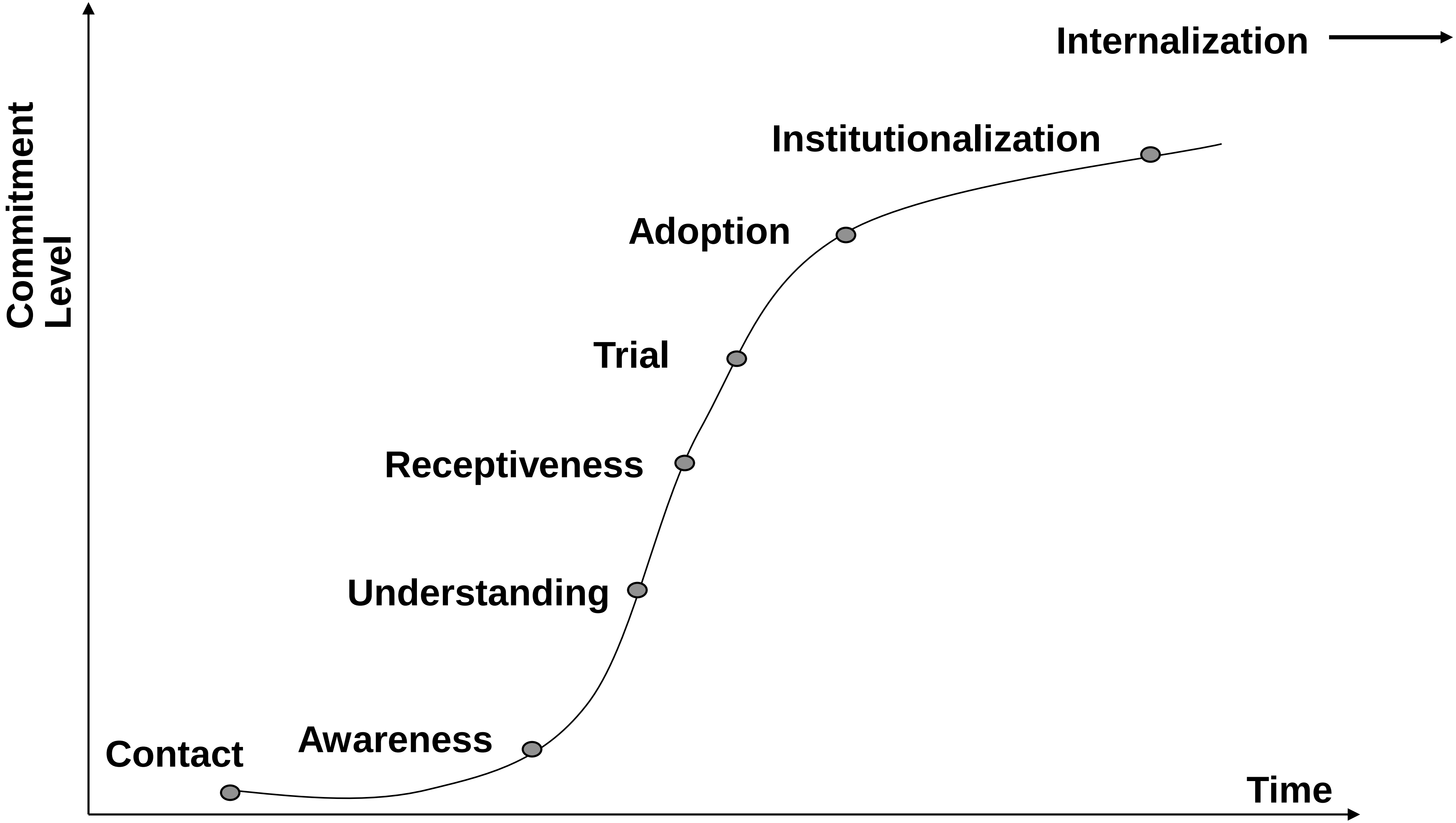
=

Behavior
Never attend another code review

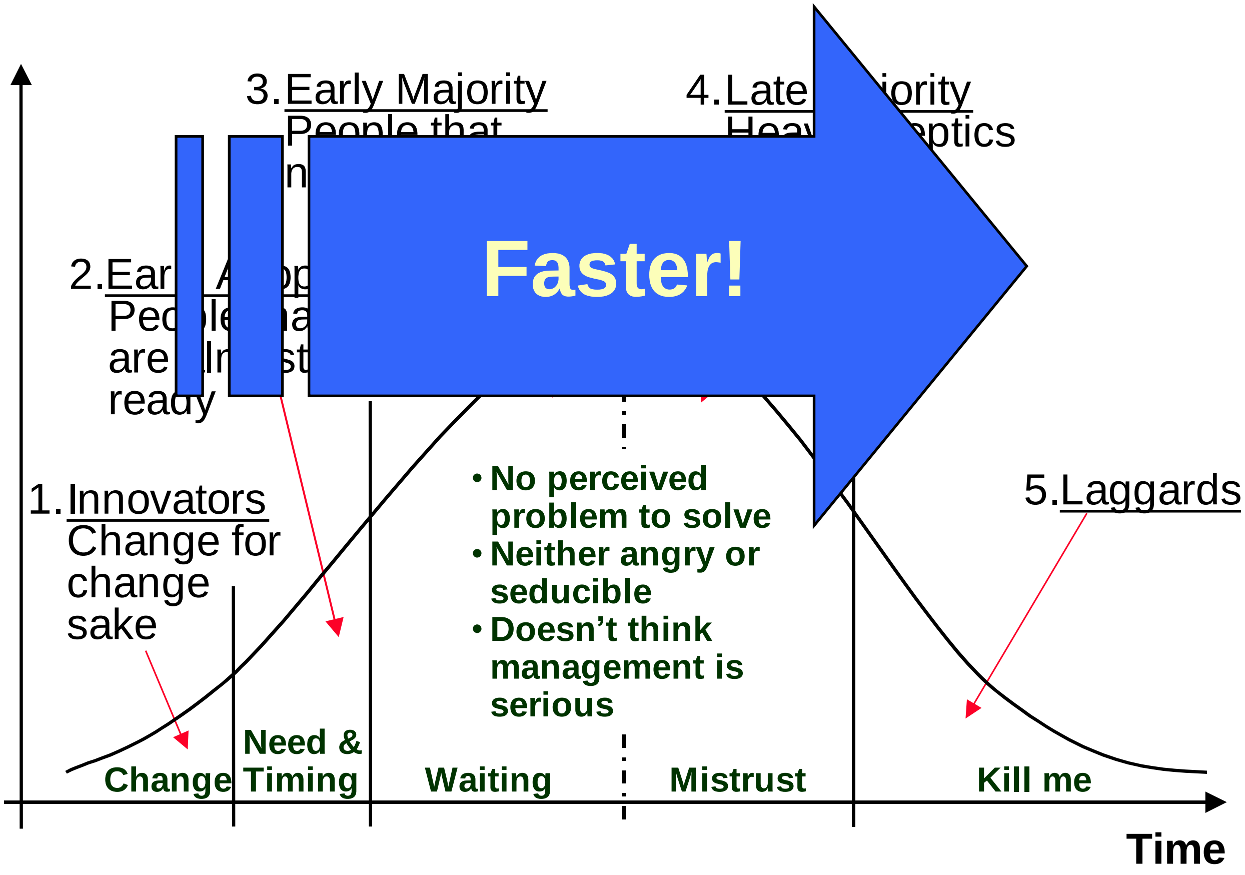
Overcoming Resistance - Beliefs - 2

- **Understand customer’s <u>values</u>**
  - What is most important to you?
  - What is most important to you about planning?
- **Understand <u>beliefs</u>**
  - What have you heard about code reviews?
  - What have your experiences been with process improvement?
- **Use discussion, new information and events to help correct any inaccurate beliefs**
  - For example:
    - ensure that the trial of a new idea is successful
    - use testimonials

Plan to Reduce Resistance



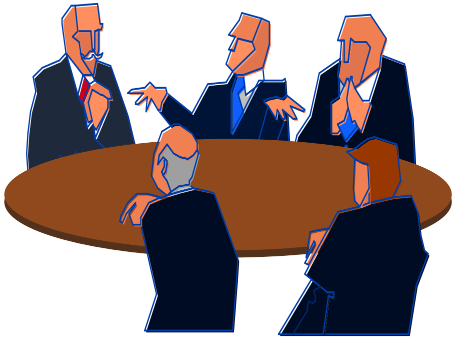
Accelerating Adoption Through Training and Consulting - 1



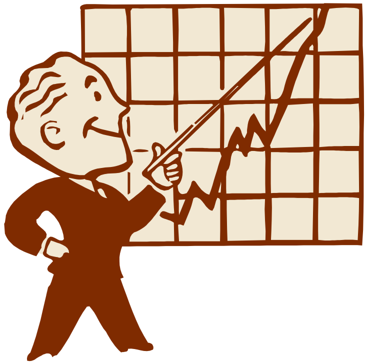
- No perceived problem to solve
- Neither angry or seducible
- Doesn't think management is serious

Accelerating Adoption Through Training and Consulting - 2

Coach



Facilitator



Presenter / teacher

- Change Agents: get out of your office and help!
- Always work on real problems
- Get other process champions to help you deploy solutions

Politician / go-between

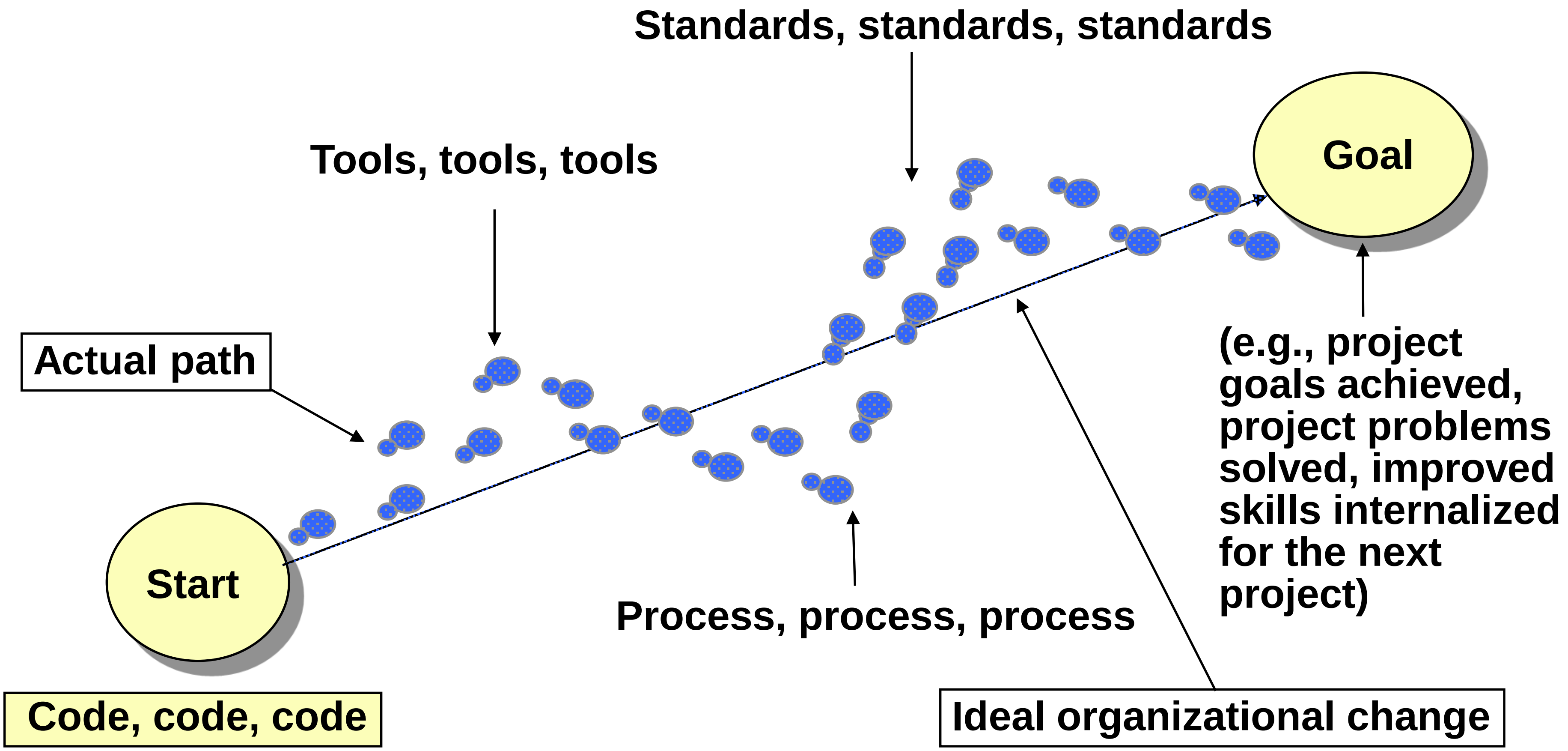
Process Champions and Train-the-trainer

- **Process Champion characteristics:**
  - Highly respected (by the target audience)
  - Some dedicated time to help
  - Able to share information practically
- **Train-the-trainer guidelines:**
  - Provide the champion with materials
  - Observe the champion train an audience
  - Keep training to small (2-4 hr), manageable topics, e.g.,
    - » **Risk management**
    - » **Inspection**
    - » **Estimation**

Implementing the Plan

- Sell Solutions Based on Needs
- Work with the Willing and Needy First
- Keep Focused on the Goals and Problems
- Align the Behaviors of Managers and Practitioners
- Avoiding a Documentation Glut

Keep Focused on the Goals and Problems - 1



Standards, standards, standards

Tools, tools, tools

Goal

Actual path

(e.g., project goals achieved, project problems solved, improved skills internalized for the next project)

Start

Process, process, process

Code, code, code

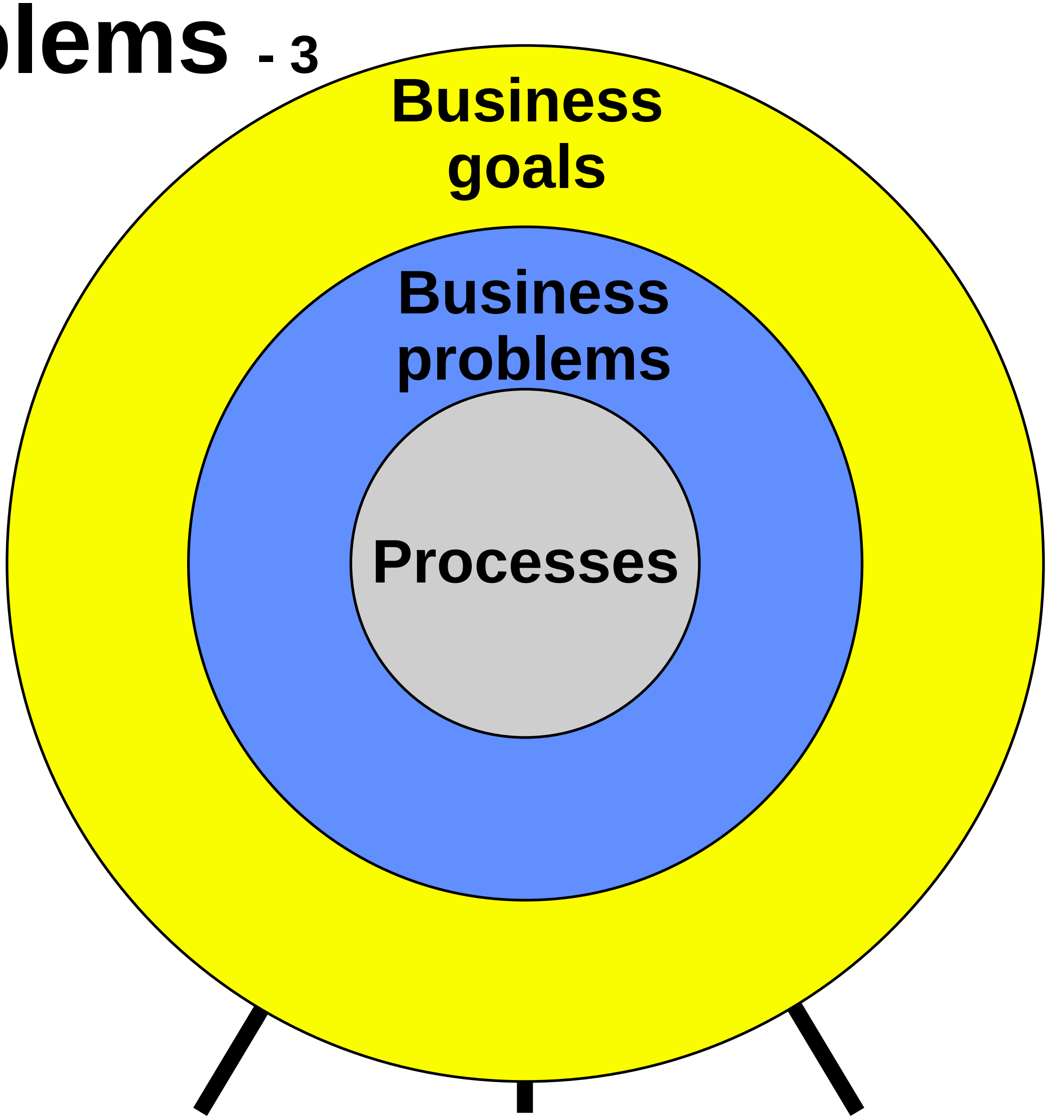
Ideal organizational change

Keep Focused on the Goals and Problems - 2

- Review improvement plans periodically during departmental and project meetings.
  - If you are not making weekly gains in your improvement program, you may be off track. Weekly gains come from fixing numerous, small project-level problems.
- Are improvement activities tied to the business goals and problems experienced by the organization?
- Are projects getting better results?
  - e.g., improved customer satisfaction, less rework, fewer surprises, fewer communication problems, meeting deadlines etc.

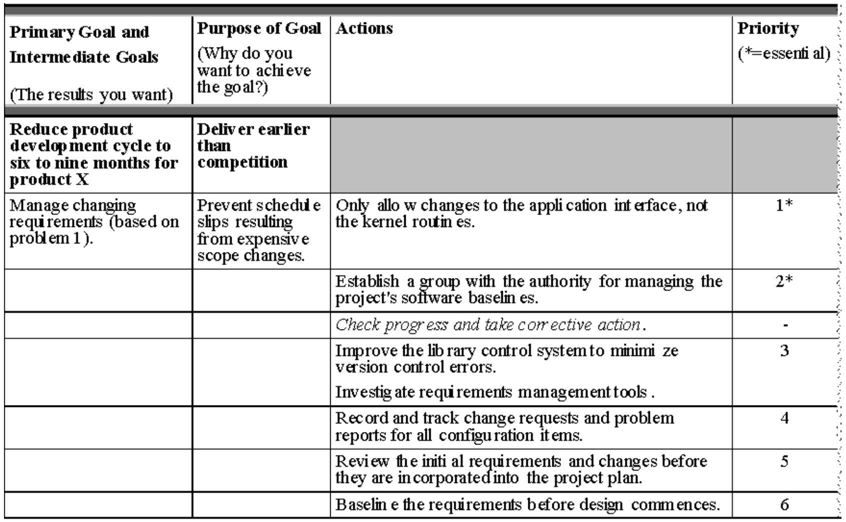
Keep Focused on the Goals and Problems - 3

- **Don't revert back to process-centric improvement**
  - One framework topic at a time
- **Practice and internalize goal-problem approach**



Keep Focused: Doing Too Much at Once - 1

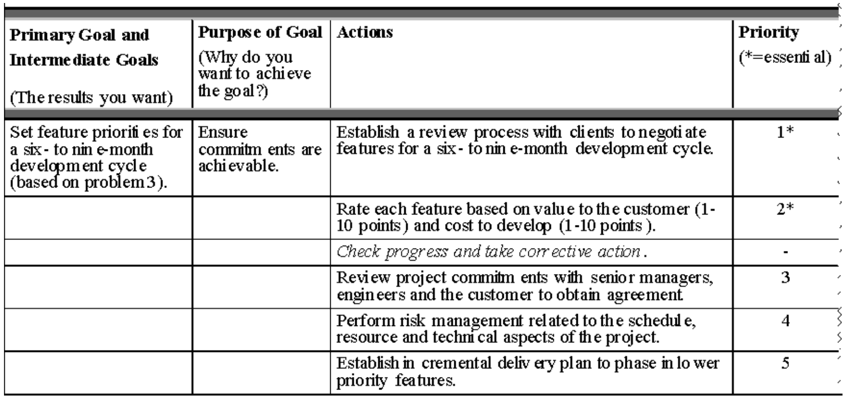
- **Monday's plan**



Primary Goal and Intermediate Goals (The results you want)	Purpose of Goal (Why do you want to achieve the goal?)	Actions	Priority (*=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*
		Establish a group with the authority for managing the project's software baselines.	2*
		Check progress and take corrective action.	-
		Improve the library control system to minimize version control errors. Investigate requirements management tools.	3
		Record and track change requests and problem reports for all configuration items.	4
		Review the initial requirements and changes before they are incorporated into the project plan.	5
		Baseline the requirements before design commences.	6

QuickTime™ and a TIFF (LZW) decompressor are needed to see this picture.

- **Tuesday's plan**



Primary Goal and Intermediate Goals (The results you want)	Purpose of Goal (Why do you want to achieve the goal?)	Actions	Priority (*=essential)
Set feature priorities for a six- to nine-month development cycle (based on problem 3).	Ensure commitments are achievable.	Establish a review process with clients to negotiate features for a six- to nine-month development cycle.	1*
		Rate each feature based on value to the customer (1-10 points) and cost to develop (1-10 points).	2*
		Check progress and take corrective action.	-
		Review project commitments with senior managers, engineers and the customer to obtain agreement	3
		Perform risk management related to the schedule, resource and technical aspects of the project.	4
		Establish incremental delivery plan to phase in lower priority features.	5

Keep Focused: Doing Too Much at Once - 2

- Success in any discipline is accomplished by focusing on a few items at a time
  - Stick to the priorities you established in your plan
- See a few improvements to completion so the organization can experience success
- Early successes provide fuel and motivation to address the remaining goals and problems

Implementing the Plan

- – Sell Solutions Based on Needs
- – Work with the Willing and Needy First
- – Keep Focused on the Goals and Problems
- – Align the Behaviors of Managers and Practitioners
- – Avoiding a Documentation Glut

What Management Can Do - 1

- **At all management levels:**
  - Place improvement in **top five priorities**
  - Communicate PI **for progress on business goals and problems**
    - » PI is not documentation for some appraiser! (Documentation captures a solution to a problem)
  - Maintain overall **department improvement plan** and track publicly
  - Ensure that improvement actions are in **current plans**
    - » Fix current project problems as we go
    - » Mitigate current project risks as we go
    - » Include time in schedules for improvement + use allocated rework time
  - Lead by **example** (PP, CM, REQM, vendor selection)

What Management Can Do - 2

- At all management levels:
  - Maintain consistency of purpose
    - » Avoid flavor-of-the-month
    - » Internalize the use of process models
    - » Understand how this practice helps us
  - Establish, track and use measures
    - » E.g., customer satisfaction, defect density, planned vs. actual time
  - Provide forums for sharing good practices

What Management Can Do - 3

Reinforcement and reward will condition a new behavior long-term. The reward must be timely and meaningful.

- **Meaningful: valuable to the person or group getting the reward.**
- **Timely: as soon as possible after the event.**
- **Don’t reward firefighting or heroism, unless that is a long-term goal!**

Examples:

- **Benefit from the improvement**
- **Recognition**
- **Increased responsibility**
- **Money**
- **Free time**

What Management Shouldn’t Do

- **At all management levels, DON’T:**
  - Manage PI as an activity **completely unrelated to running the business** - a documentation exercise!
  - Make the **EPG solely responsible** for achieving CMMI Level N
  - **Undermine** improvements
    - » e.g., “Forget CM, E-mail the product now!”
  - **Overcomplicate** solutions
    - » 25 metrics
    - » Globally available historical database, 15 countries
    - » Every CMMI practice uniquely met, instead of merging them together
    - » Documentation = 20 pages per process

Implementing the Plan

- – Sell Solutions Based on Needs
- – Work with the Willing and Needy First
- – Keep Focused on the Goals and Problems
- – Align the Behaviors of Managers and Practitioners
- – Avoiding a Documentation Glut

Avoiding a Documentation Glut

Documentation Key Issues

1. Focusing on the organization’s needs
2. Keeping processes concise
3. Knowing when you are in trouble
4. Knowing if you are meeting the intent of the framework (e.g., CMMI process areas)

1. Focus on the Organization’s Needs

QuickTime™ and a
TIFF (LZW) decompressor
are needed to see this picture.

- **Keep process documentation concise by focusing it on specific needs (e.g., business goals and problems)**
- **Begin with a simple version of the process. When the need is addressed, stop**
  - Refine further when the process no longer meets the need

Focus on the Organization’s Needs

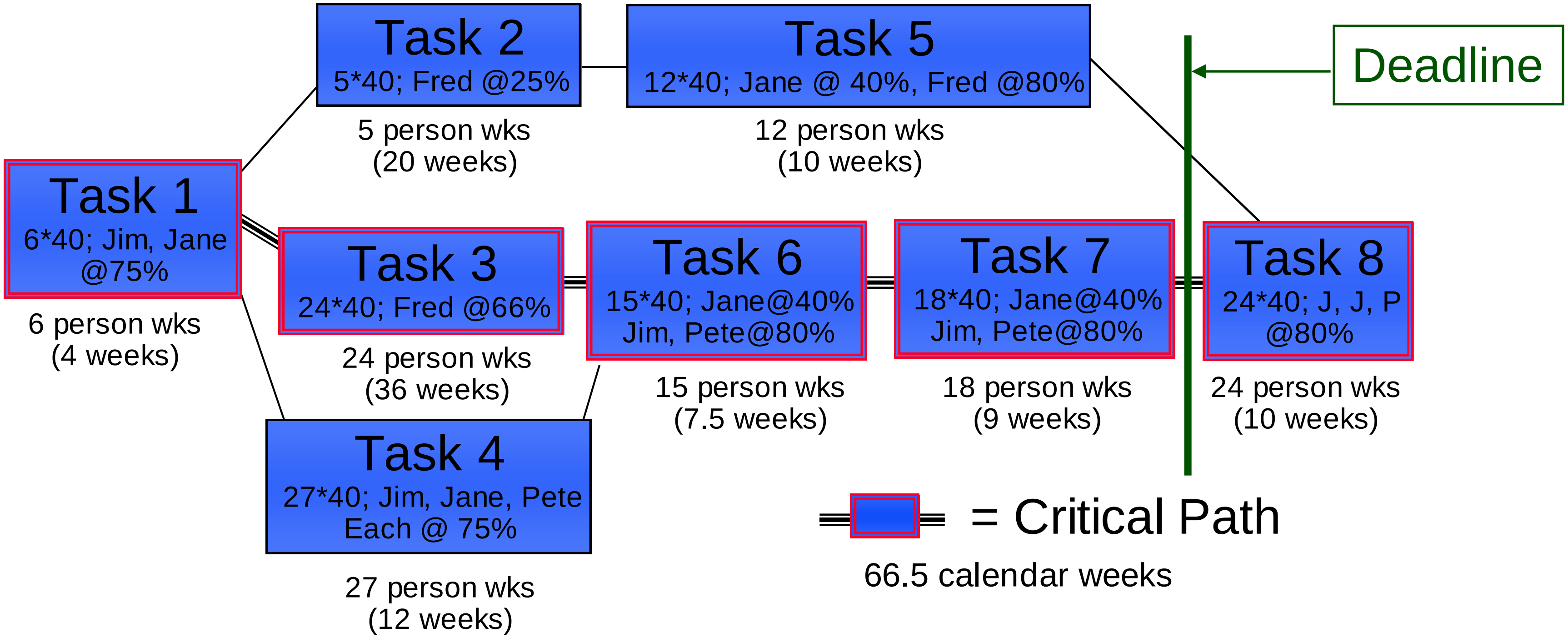
Example

Project Needs	SEI CMMI Practices That Would Help
Changing requirements.	**Level 2: PP - Specific Practice 2.1** Establish and maintain the projectÕs budget and schedule. **Level 2: PP - Specific Practice 3.2** Reconcile the project plan to reflect available and estimated resources.
The poor quality of incoming code from other groups.	
We routinely over commit.	
Inadequate availability of test equipment.	
Too many features are required for the 6- to 9-month development cycle.	
Difficult to find defects early.	

Use the need(s) to scope the process

Example Process for Schedule Creation

(Level 2: PP - Specific Practice 2.1 <Schedule>)



1. Determine task dependencies.
2. Add task EFFORT estimates.
3. Add resources - people, equipment, resource assumptions.
4. Add resource availability - %allocation, calendar days out.

Example Process for Reconciling Commitments

(Level 2: PP - Specific Practice 3.2)

Step 1: Project team determines high-level product needs (or scope of work), from customer and marketing input

Step 2: Project team develops an initial project plan and estimates to determine what is feasible

Step 3: Project team meets with management, marketing, customers and related groups to determine whether:

- the change or product is feasible
- there is agreement to the resource, cost and schedule estimates
- the risk is acceptable

Step 4: A commitment is made OR further negotiation is held

2. Keep Processes Concise - 1

- **Always consider 1 page (small) for each process or sub process!**
  - Refine what you have defined, don’t necessarily add more
- **A Defined/Managed Process can be the instructions embedded in a work product template; e.g.,**
  - The template for an CM, QA or project plan
- **A standing agenda can be the process for a project review**
  - With instructions for use

CM Plan Template

1. List Configuration Items

   x, y, z

2. Establish File Naming Convention

   File-x<n>.doc

3. Establish Baseline File Structure

   ~~~~~~

4. ~~~~
   ~~~~~~

Example Milestone Review Process

(Level 2: Project Monitoring & Control - Specific Practice 1.7)

- **For the last period:**
  - The **original plan**
  - **Accomplishments**
  - The **critical path** of the project
  - **High-risk areas** that need attention (top 2-3)
  - **Problems** that are impacting quality, cost and the schedule
  - **Status of action items** (open and closed)
- **For the next period:**
  - **The plan**

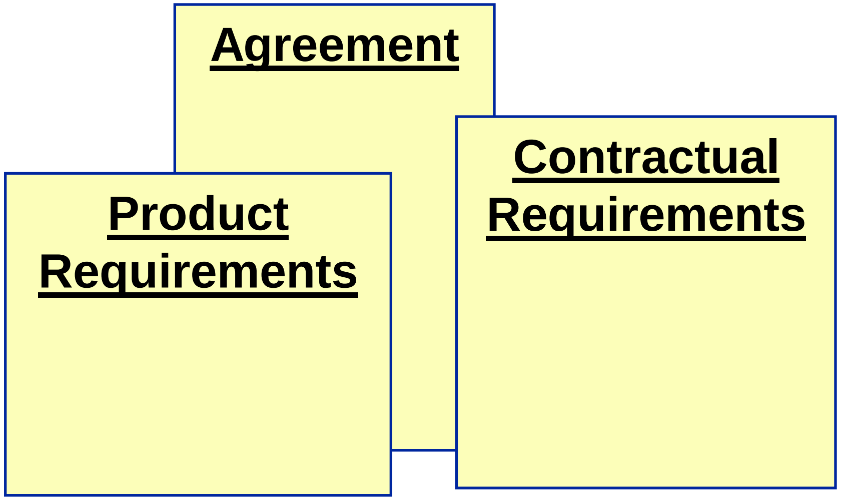
Instructions for use:

~~~~~~~~~~~~~~

~~~~~~~~~~~~~~

Keep Processes Concise - 2

- **Merge duplicate work products, e.g.,**
  - Just because they might be listed separately in the CMMI does not mean that you can't merge them

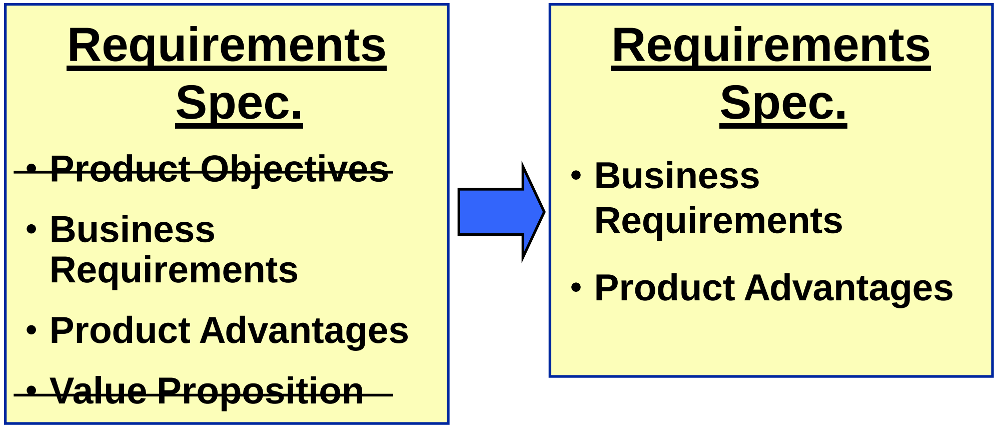


Agreement

Product Requirements

Contractual Requirements

- **Remove redundancy in templates**



Requirements Spec.

- ~~**Product Objectives**~~
- **Business Requirements**
- **Product Advantages**
- ~~**Value Proposition**~~

Requirements Spec.

- **Business Requirements**
- **Product Advantages**

Keep Processes Concise - 3

- Don't have separate QA checklists that repeat the original process. Use the original process as the checklist.
  - Add specific QA pointers and guidelines.

- Look for reuse in your CMMI implementation, e.g.,

CMMI Practice	This CMMI Practice Can be Used Here Too
Level 2: PP - Specific Practice 1.1 Establish a top-level work breakdown structure (WBS) to estimate the scope of the project	**GP 2.2 in all Process Areas** Plan the Process + use PA process description
Level 3: VER - Specific Practice 2.2 Conduct Peer Reviews	**Level 3: TS - Generic Practice 2.9** Evaluate adherence of the technical solution process

VER = Verification, TS = Technical Solution

Keep Processes Concise - 4

- **Consider one representation**
  - e.g., PowerPoint can be printed, shared and presented
- **Embed tailoring guidelines**
  - A separate document can be difficult to find and update
- **Have one policy**
  - e.g., “Perform the life cycle” Follow the tailoring guidelines in the lifecycle

→

<u>Estimation Process</u>

Step 5: Use the historical database to verify the estimate for each task

<u>Purpose</u>: To search the organization's historical data to see if a similar task (or group of tasks) exists.

***<u>Tailoring guideline</u>*:** This step should be performed whenever applicable data exists. It can be discarded when a new language or technology is being used.

<u>Risk if omitted</u>: Failure to use the database could result in significant oversight about schedule estimates, and could lead to a loss in revenue.

<u>Minimum requirement</u>: Data that exists, but is not considered applicable for the current estimate, must be reviewed with one other manager to verify non-applicability.

3. Know When You are in Trouble

- Project team members create process and project documentation to please an appraiser or auditor

QuickTime™ and a
Photo - JPEG decompressor
are needed to see this picture.

- It has been 6 months and still the process is not ready to use

QuickTime™ and a
TIFF (LZW) decompressor
are needed to see this picture.

- Project managers “study” the documentation in preparation for the appraisal

- The ink refuses to dry, and the appraisal interview is about to start!

4. Know if You are Meeting the Intent of the framework

(e.g., CMMI Process Areas)

- **The problems related to those Process Areas have been solved and the solutions are captured in the processes.**
- **Project and process documents are used to run the project and the business.**
  - The practices within the CMMI have been institutionalized. The process "lives."
  - No "extra paperwork".
- **The processes have "institutionalized" characteristics.**
  - E.g., documented, planned, resourced, trained, someone assigned, under control, meet needs, monitored.

Documentation Summary

Process documentation is:

- Only a small part of process improvement
- A method of capturing and sharing engineering and management practices

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**

Exercise: Implementing the Plan

1. **Make notes on which project members are the innovators and early adopters, and what they are ready to adopt. Refine your current improvement action plan using this information. What other interviews do you need to conduct?**
   - Also, consider the late majority (skeptics) and laggards.
2. **Review your improvement plan: Is it aimed at deploying solutions in small pieces based on project needs and priorities?**
   - For example, a complete project planning process can be broken into estimation, negotiation, risk identification, and scheduling.
3. **Develop a plan to interview some managers and find one who can use some of the techniques you are advocating, such as:**
   - Estimating, planning or peer reviews.
4. **Identify someone who can help deploy well-tried improvements in parallel with your efforts.**

Result:

Action Items

```
1. ~~~~~~~~~~
~~~~~~~~~~~~~
2. ~~~~~~~~~~
~~~~~~~~~~~~~
~~~~~~~~~~~~~
3. ~~~~~~~~~~
~~~~~~~~~~~~~
~~~~~~~~~~~~~
4. ~~~~~~
~~~~~~~~~
```

Checking Progress

> “You can design a measurement system for any conclusion you wish to draw.”
>
> —Gerald Weinberg, *Quality Software Management*

The Purpose of Checking

- **Checking progress:**
  - lets you know how well your improvement program is going
  - provides visibility to detect problems early
  - gives you data to make your future plans more effective
- **Corrective action consists of:**
  - mid-course changes based on results and lessons learned from the planning and implementation phases

Goal-Question-Metric Approach - 1

Goal	Questions	Metrics
Meet all our cost and schedule commitments.	Are we spending the planned number of hours on the project to complete it? Are we hitting our milestones?	Planned versus actual effort for each project. The number of days each milestone is early or late.
Deliver product X by mm/dd/yy.	Are we spending the planned number of hours on the project to complete it? Are we hitting our milestones?	Planned versus actual effort for each project milestone. The number of days each milestone is early or late.
Reduce rework to less than 20 percent of total project effort.	How much time do we spend on rework now? How does this compare with our development time and are we improving?	Percentage of project time spent on rework.
	How many defects do we have in the product during design and coding?	Defect density: Number of defects found per unit size of work product (e.g., number of pages of design, number of lines of code).

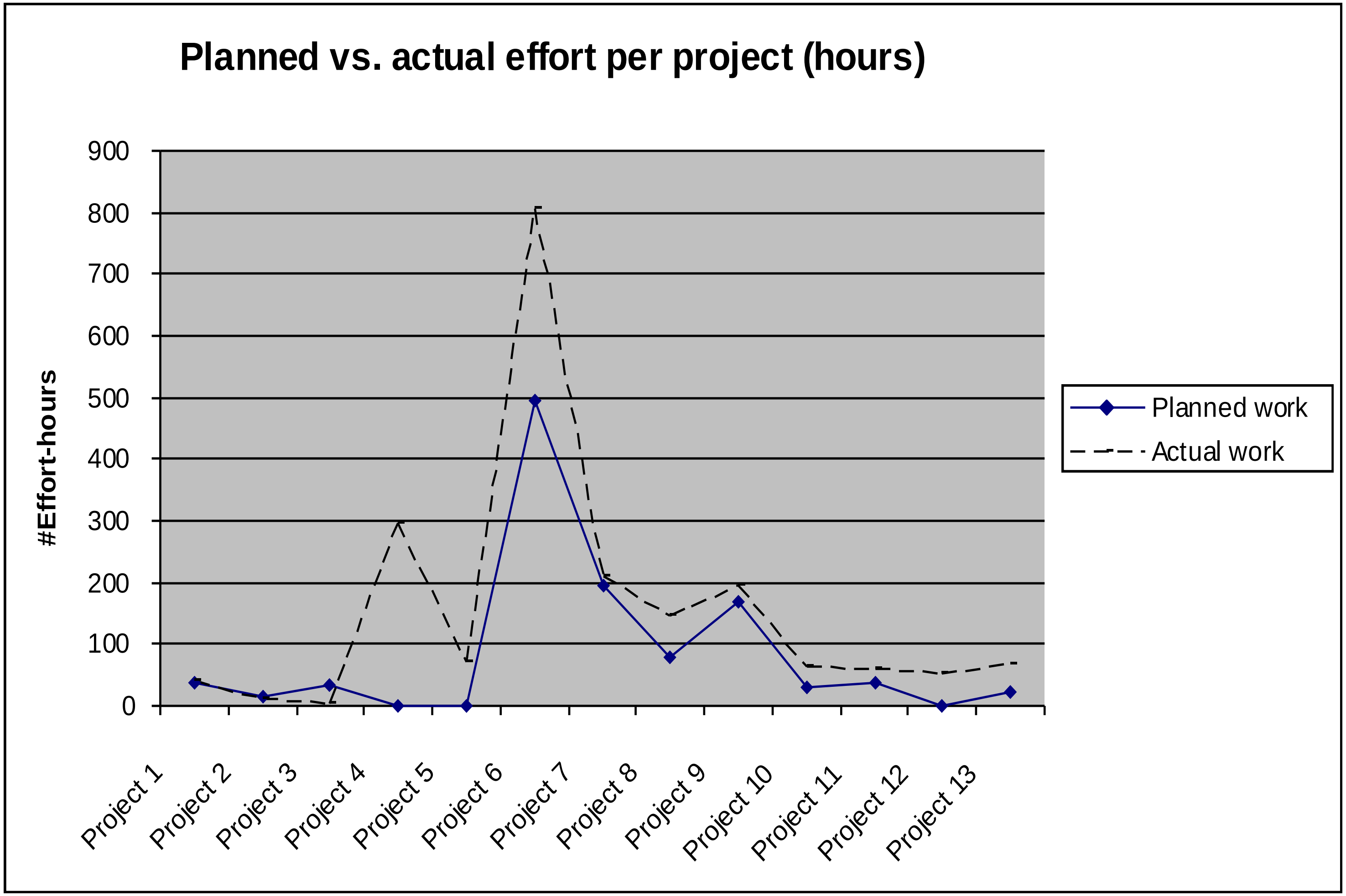
Goal-Question-Metric Approach - 2

Goal	Questions	Metrics
Achieve customer rating of 9/10 on product evaluation form.	How satisfied are they now? Are we improving?	Annual customer satisfaction survey.
Keep profits at 15 percent (and costs at the same as last year).	What is our profit? Is it getting better or worse?	Annual net profit.

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?

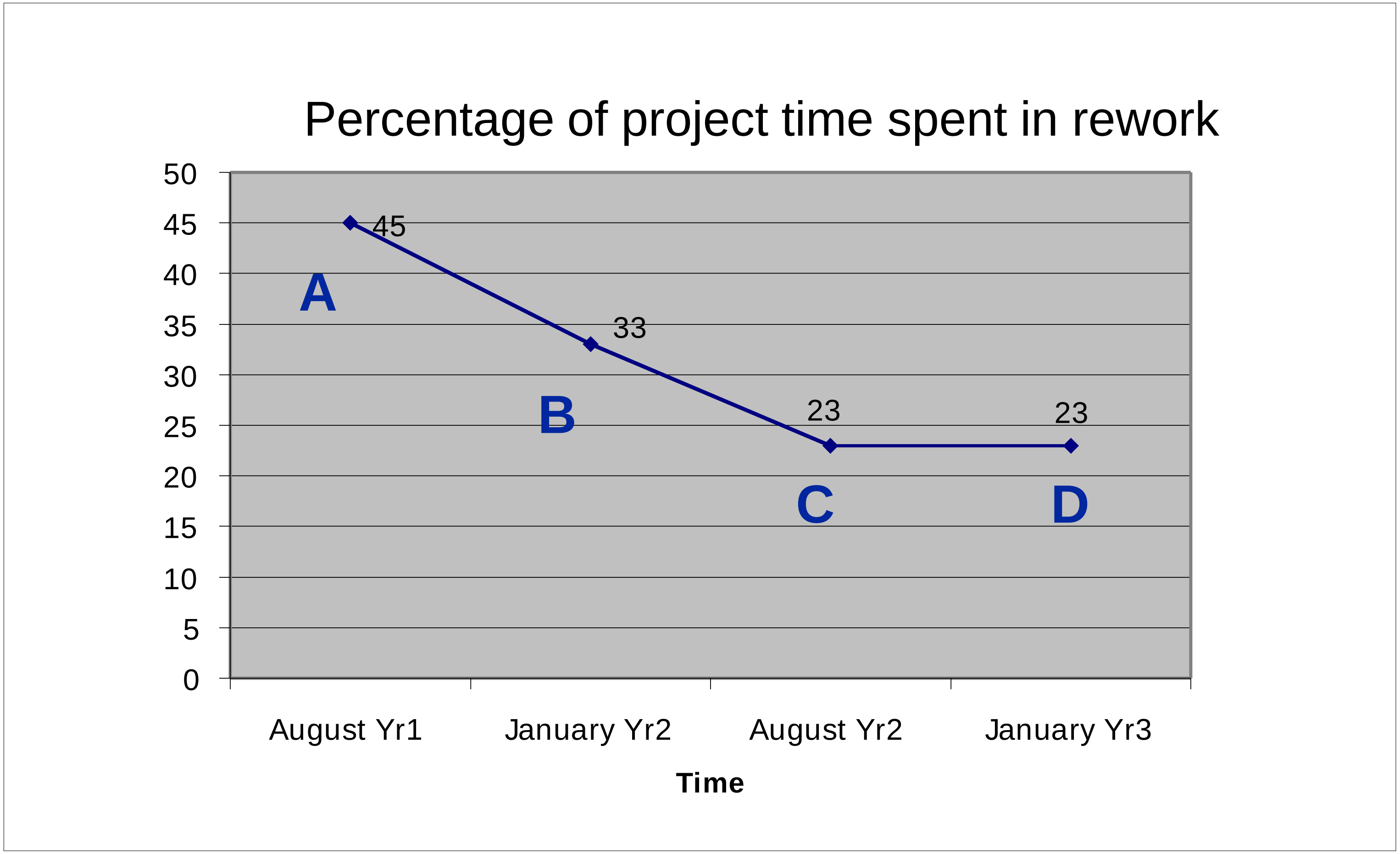
Goal: Meet all Our Cost and Schedule Commitments



Observations and Corrective Actions

- **Tracking the data for each project showed:**
  - a trend of consistently underestimating the number of hours needed
  - that although the group met the majority of their deadlines, the hours expended to do so were causing some financial loss
- **Corrective actions:**
  - develop a spreadsheet of historical information and use it when estimating new projects
  - use the Wide Band Delphi technique for deriving estimates
  - share the data with the sales staff to develop a joint understanding of how to bid on future projects

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

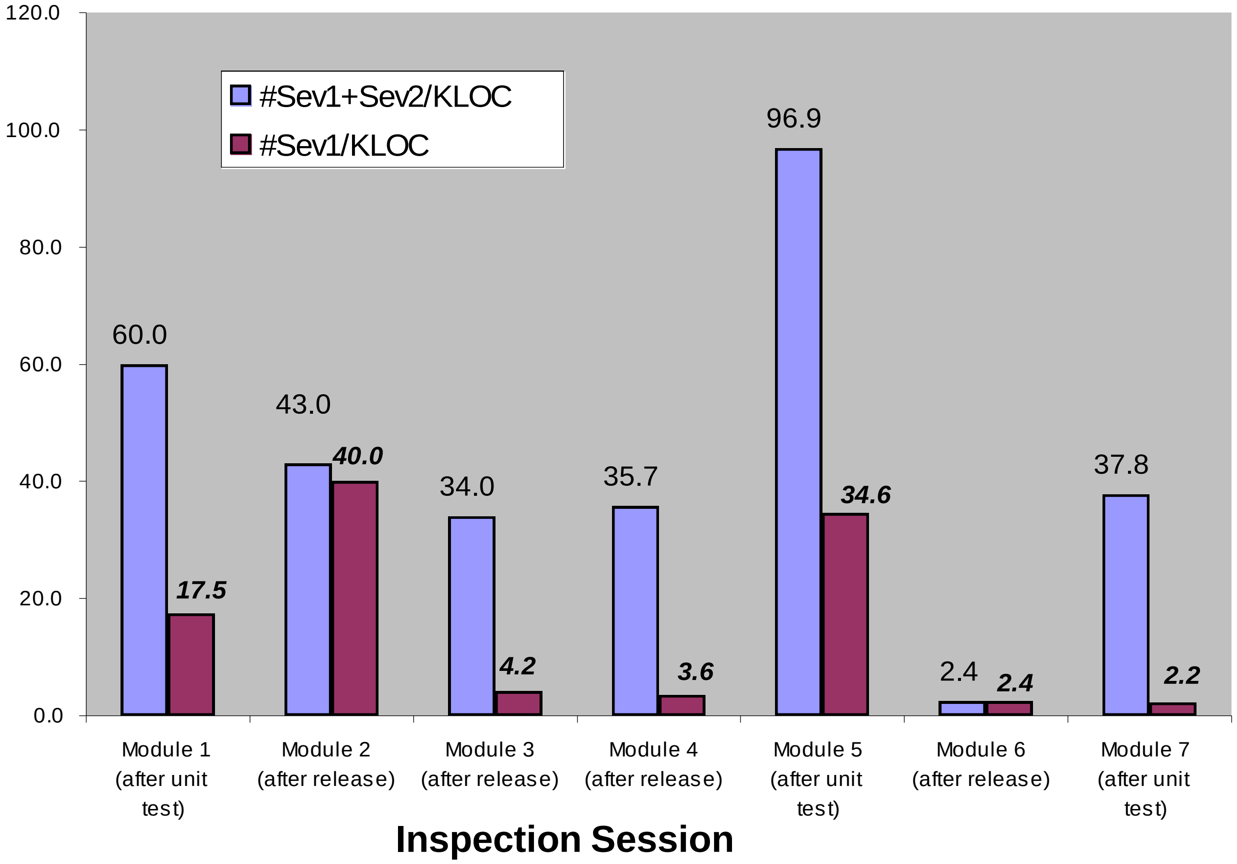


Observations and Corrective Actions

- **The graph showed:**
  - a trend of improvement in how engineering time was used
  - that further improvements were necessary to achieve the goal
- **Corrective actions:**
  - A⇨B: effort estimation, risk management, schedule creation, project tracking, inspection of design documents
  - B⇨C: inspecting code and requirements documents, formal CM, improved testing, process assurance, post-project sessions on lessons learned
  - D⇨:plan to adopt use cases and design process

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

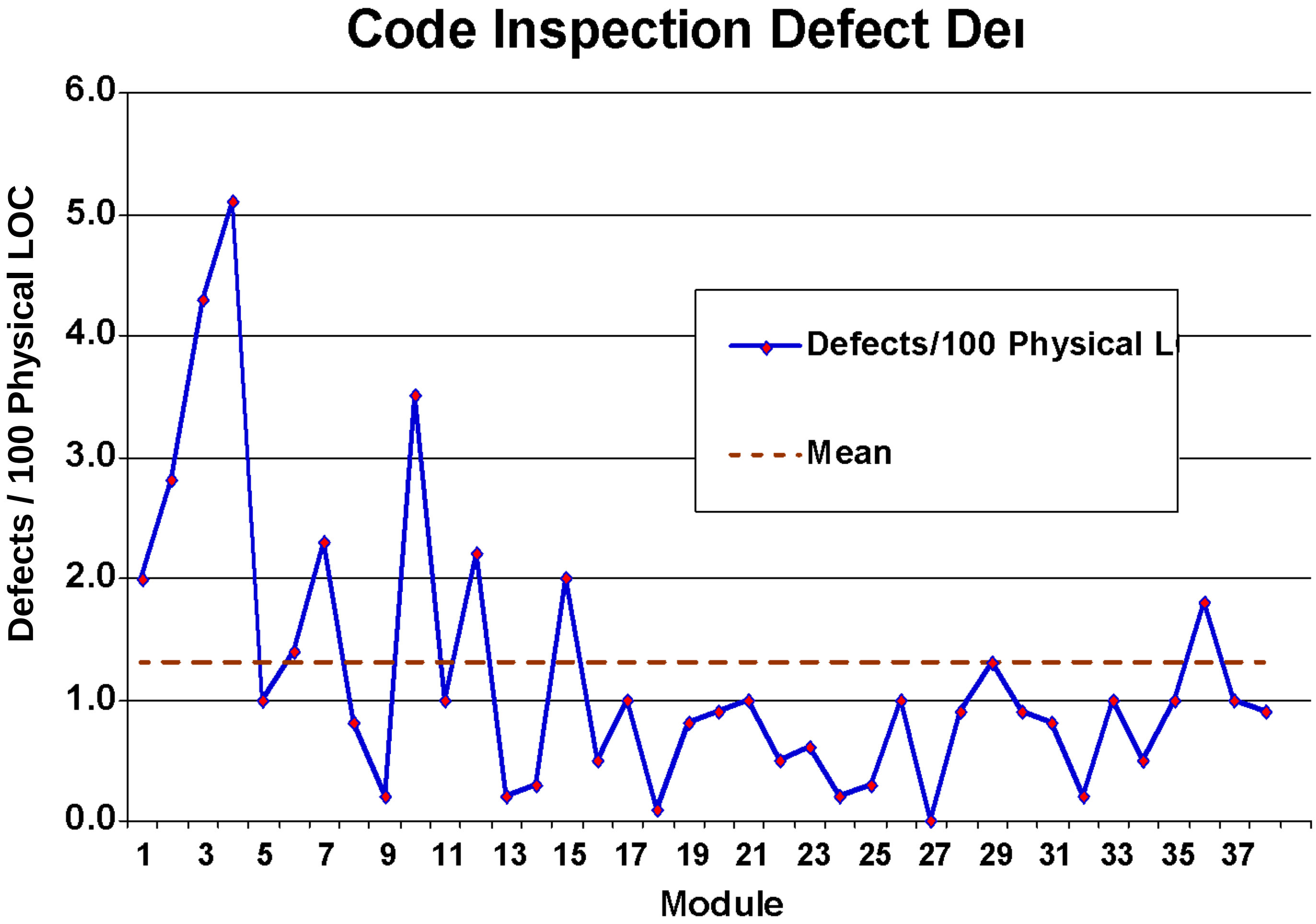
Java/C++ Inspections – Severity 1 + Severity 2 Defects per Thousands of Lines of Code



Observations and Corrective Actions

- **Defect density of released and tested software is extremely high**
  - a cause of chaos and 70% rework
- **Corrective actions:**
  - inspect a larger portion of current code base
  - develop common errors checklist to capture coding mistakes

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**

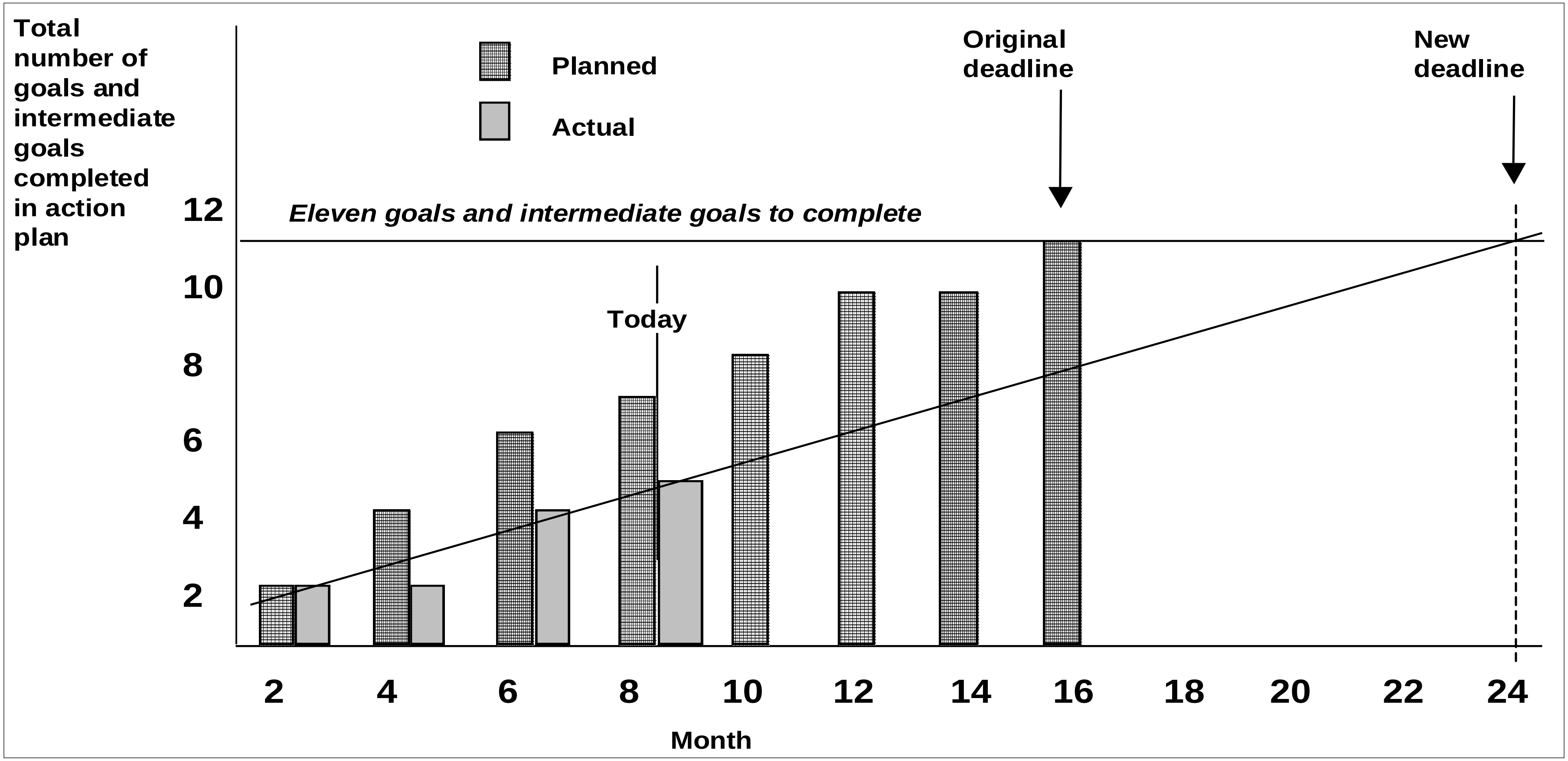
Observations and Corrective Actions

- Defect density of released and tested software is extremely low
  - only 1-3% of project effort spent on rework
- Corrective actions:
  - continue to inspect all work products

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?

Are we Making Progress on Our Improvement Plan?



Trend diagram tracking goal and intermediate goal completion

Observations and Corrective Actions

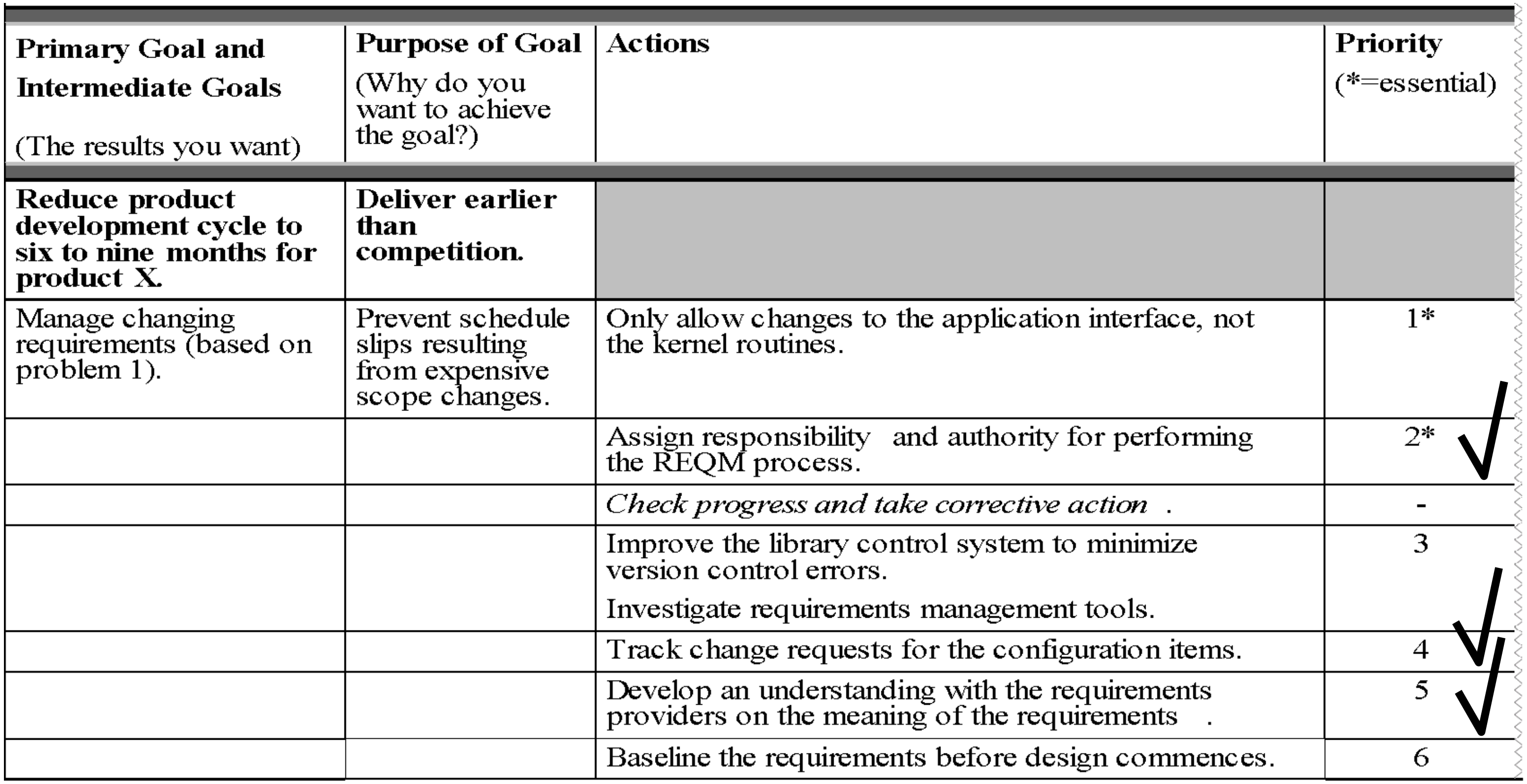
- **No way of meeting initial deadline of 16 months**
- **Corrective actions:**
  - revise completion date to 24 months
  - dedicate more time for improvements
  - use successes from early adopters to motivate others
  - ensure that the solutions we are developing do not exist somewhere else

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?

Are We Making Progress on the Improvement Framework? - 1

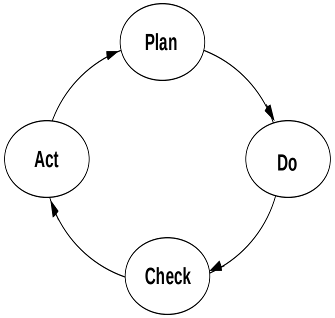
Method 1: Count actions that are from the framework



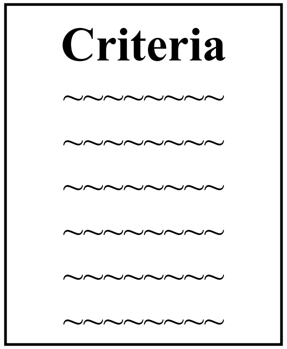
Primary Goal and Intermediate Goals (The results you want)	Purpose of Goal (Why do you want to achieve the goal?)	Actions	Priority (*=essential)
Reduce product development cycle to six to nine months for product X.	**Deliver earlier than competition.**		
Manage changing requirements (based on problem 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. 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*



Mini-assessment



Purpose:

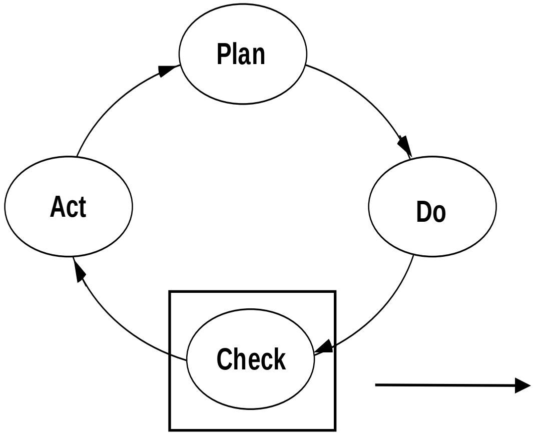
- To **evaluate improvement progress** and make necessary adjustments

Method:

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

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

Mini-assessment Process



- Plan the assessment
- Meet with interviewees to explain what will be checked, how and why (answer questions and concerns)
- Perform the mini-assessment (interviewing with questionnaire)
- Communicate the results (organization summary)
- Debrief the mini-assessment process to obtain feedback and buy-in from the organization
- Improve the questionnaire (emphasize intent and remove ambiguity)
- Take corrective action

Mini-assessment Questions

Sample mini-assessment questions

- **Describe how your team**
  - Performs inspections or walkthroughs for key work products (such as code, design, test cases, and test plans)
  - Performs black-box testing
  - Performs white-box testing
  - Performs version control of all significant work products (from plans to code)
- **Do you have adequate computer network stability (compared with the problem reported in the last assessment)?**

Mini-assessment Findings

Project A: Strengths and areas for improvement

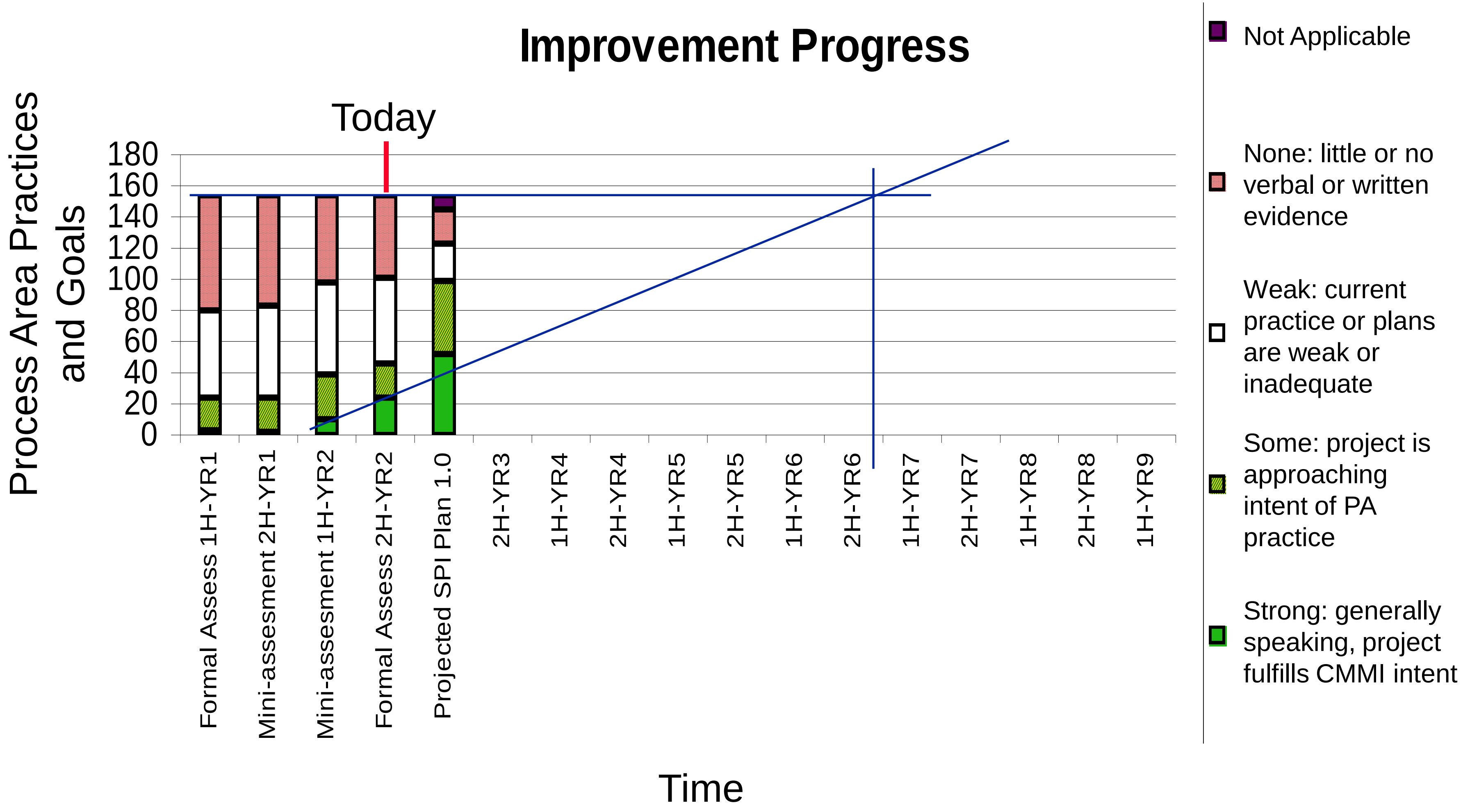
Strengths

- Inspections are performed on requirements and code.
- Black-box testing is performed against the requirements.
- White-box testing is performed on critical code.
- Work products are under configuration management (in other words, project plans, requirements, code, test plans, and test cases).

Areas for improvement

- Computer network stability has not changed since it was reported in the last assessment.
- Project plans for projects larger than three months would benefit from inspection.
- Test plans would benefit from inspection to reduce the amount of redundancy in the test approach.

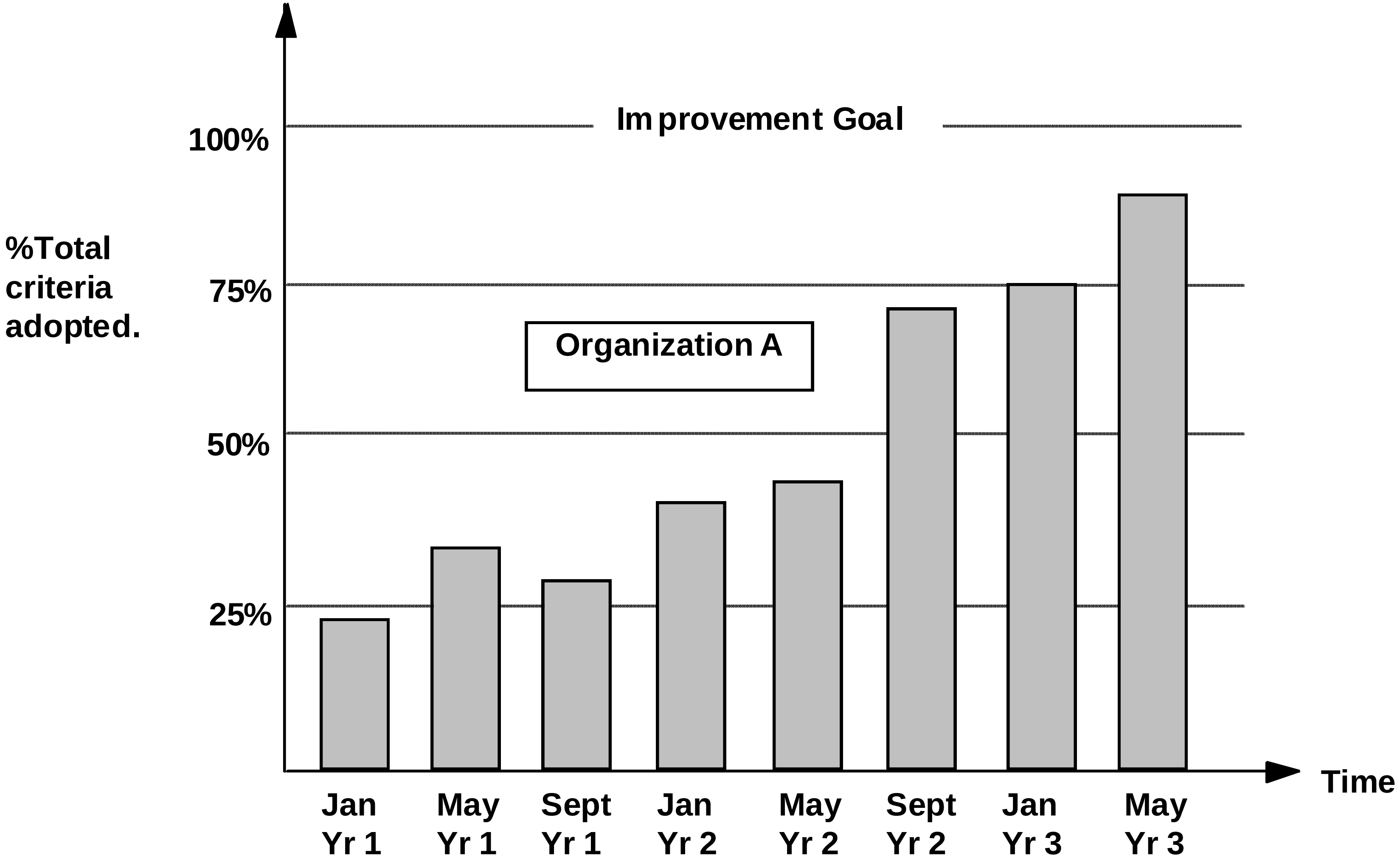
Example Mini-assessment Data - 1



Observations and Corrective Actions - 1

- **Little progress**
- **Will likely be Level 2 by second-half of year 6**
- **Corrective actions:**
  - – Report mini-assessment data (overall score and trend) to CEO and division heads - get executive visibility
  - – Replan improvement effort for each project (goals and problems)
  - – Task new engineering manager with pulling Level 2 practices into each project using bi-weekly project reviews and EPG assistance

Example Mini-assessment Data - 2



Observations and Corrective Actions - 2

- Progress is good
- Large jump between May and Sept (year 2) due to adoption of change management process, build process, and process assurance
- Corrective actions:
  - Adopt remaining Level 2 practices based on current project problems

Rules of Thumb for Measuring Adoption - 1

- **Always be careful when making measurements:**
  - **what you measure might change** when you measure it
  - always keep in mind the **quality** of the answer; a YES should be followed with:
    - » **how well does that work?**
    - » **does it help you?**
    - » **how often do you do that?**
    - » **do you do this in a crisis?**

Rules of Thumb for Measuring Adoption - 2

- **Use balanced metrics - don't just have a goal:**
  - "Adopt 50 percent of all Level 2 practices by December"
- **It may be tempting for a project team to implement the easiest 50 percent of the elements in the framework**
- **Balance this with a metric that tracks progress toward a business goal, e.g.,**
  - "Reduce defects reported from the field by 30 percent"
  - "Ensure product deliveries are no more than 15 percent late"

Tracking and observing multiple indicators show whether progress is being made on issues that impact the business

Rules of Thumb for Measuring Adoption - 3

- The goal is to establish whether the practice is a common and beneficial behavior of the project team
- Measurements take time to become refined, accurate, useful and effective
- Measurement results MUST be treated with care - don’t attribute project names
- The intent of the measure must be explicitly stated to explain how the results will be used

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?

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. 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. 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. 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. CA = adopt the goal-problem approach.	**Planning**
Using the same communication technique as everyone else allows the message to be lost. CA = use bright pink 8.5 x 11-inch cards & pizza lunches.	**Implementing**
Allowing private data to become public sets perilous expectations. CA = brief management on new metrics policy.	**Planning**
Be careful of what information you ask for! [Process Assets Library] 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. 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

Exercise: Checking Progress

1. Pick 1-3 goals and use the GQM approach to determine the metrics you need to track progress toward this goal
2. If your improvement program is already well underway, develop a plan to conduct a mini-assessment for one or more projects to assess current progress and next steps
3. Discuss lessons learned regarding your improvement program with some of the program's participants

Result:

Action Items

1. ~~~~~~~~~~
~~~~~~~~~~~~~
2. ~~~~~~~~~~
~~~~~~~~~~~~~
~~~~~~~~~~~~~
3. ~~~~~~~~~~
~~~~~~~~~~~~~
~~~~~~~~~~~~~
~~~~~~~~~~~~~

Blank slide

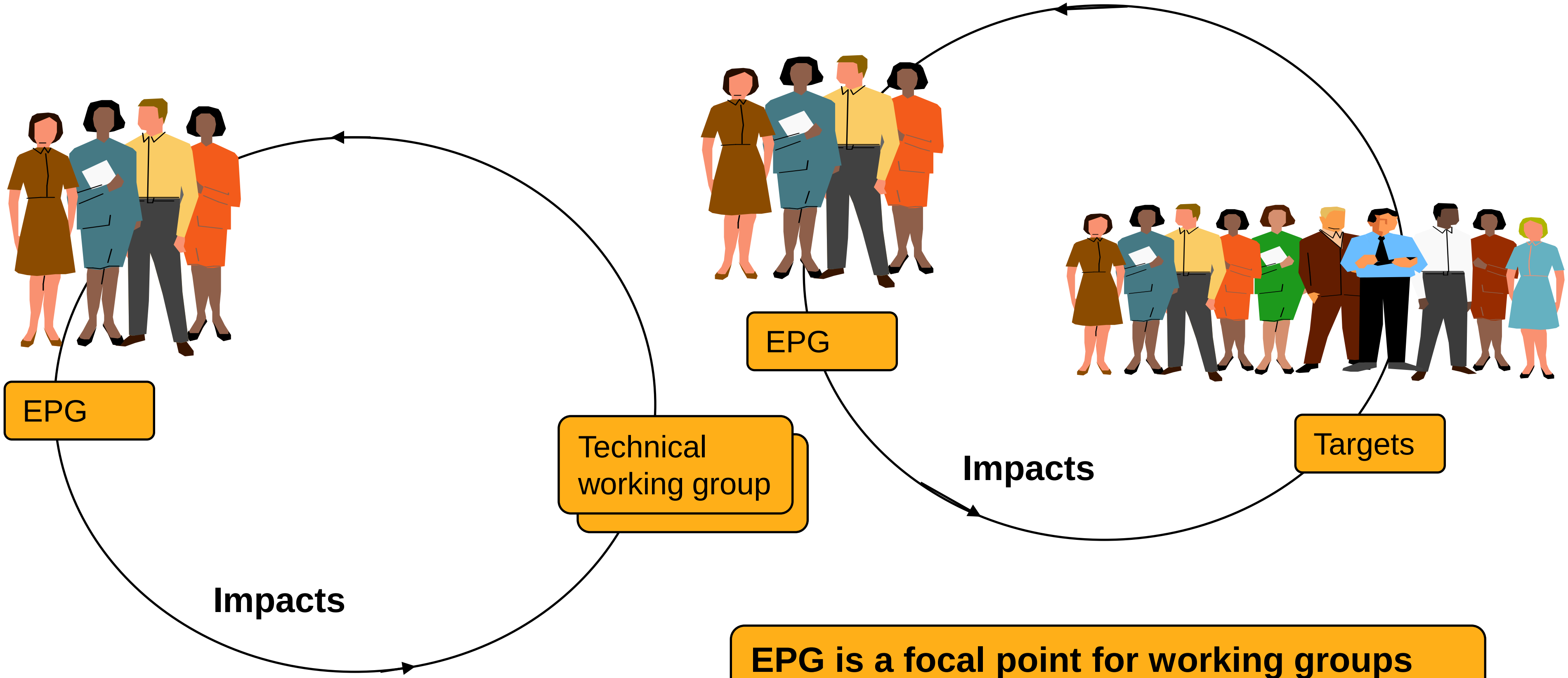
Appendix

Engineering Process Group

A means of implementing a successful process improvement program

A Engineering Process Group is a group chartered to facilitate engineering process improvement within an organization.
It helps the organization determine areas for improvement, plan the improvement effort and implement it.

Two Parallel Activities



EPG

Technical working group

Impacts

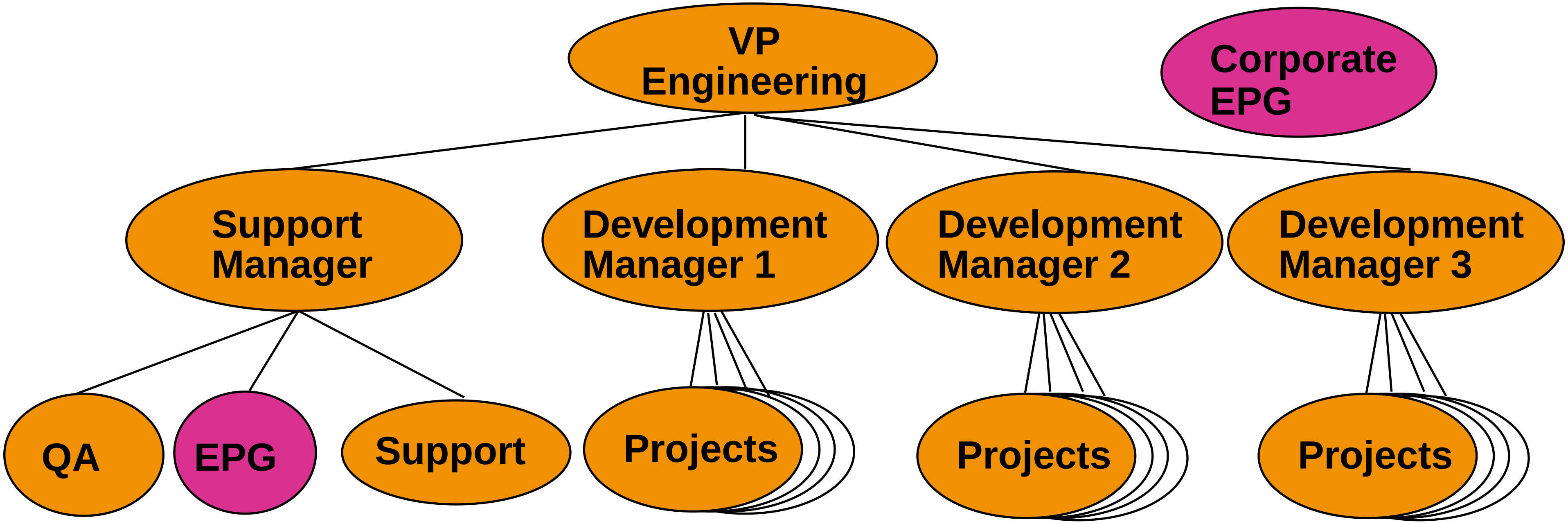
EPG

Impacts

Targets

EPG is a focal point for working groups and helps targets adopt new practices

EPG Position



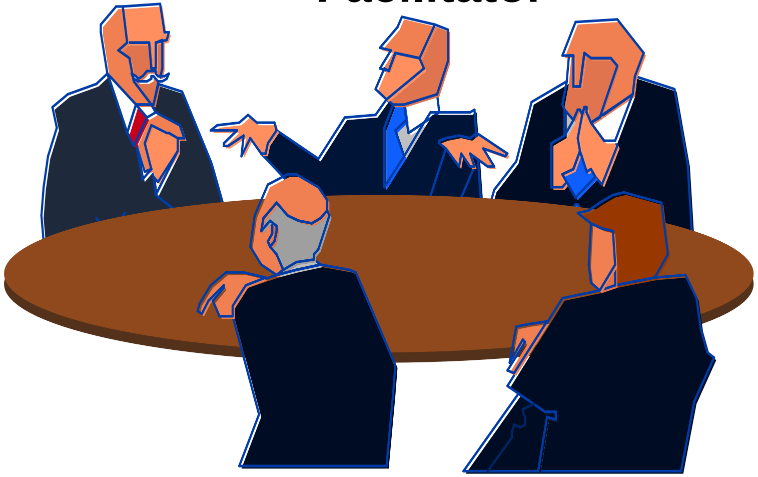
- Not part of QA. Not tied to any one project.
- Geographical and cultural dispersion.
- Career path: may be rotational (2 years), good for project leaders and managers.
- Must be seen as a respected support function to the projects.
- Champions often become EPG members. Others must be recruited. Do not accept the unwanted, idle, or disliked!

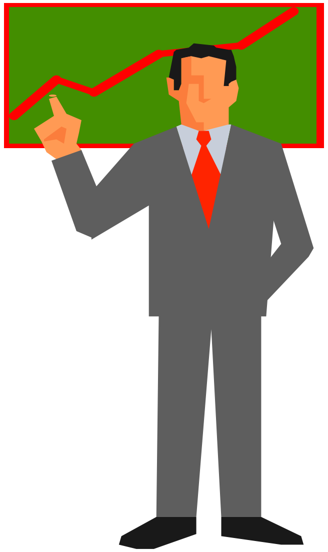
EPG Roles

Assessor

Coach

Facilitator



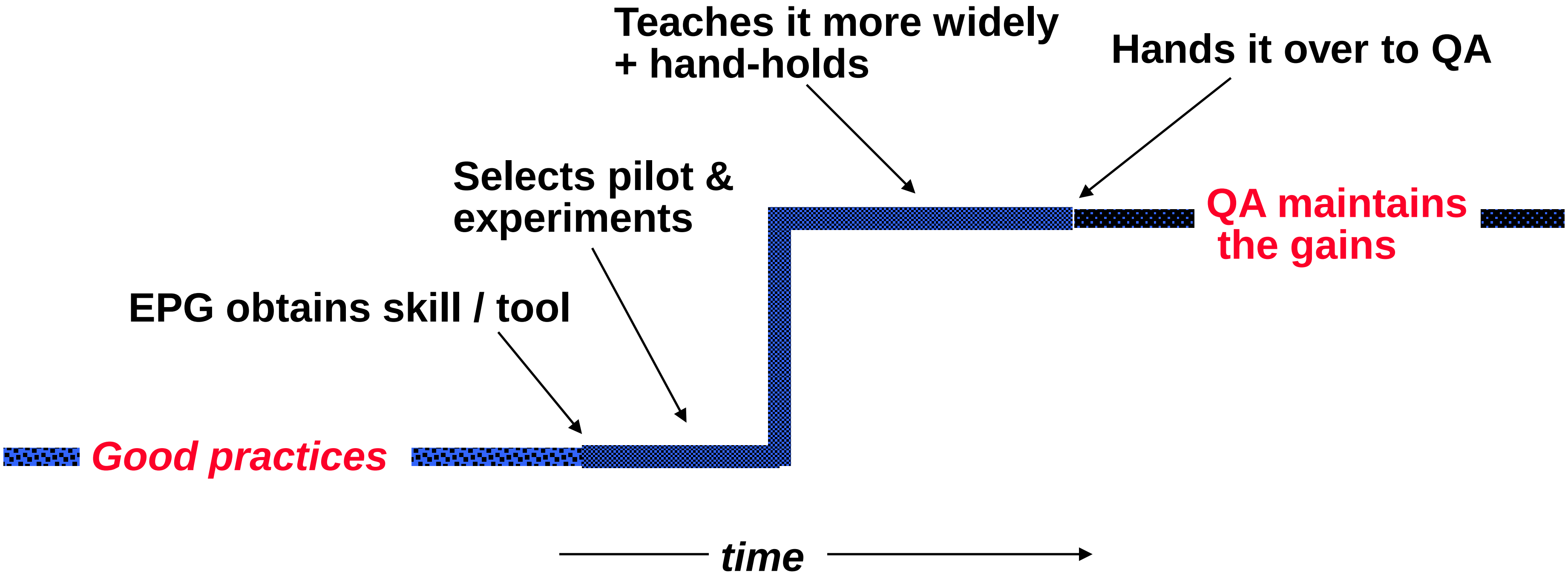


Presenter / teacher

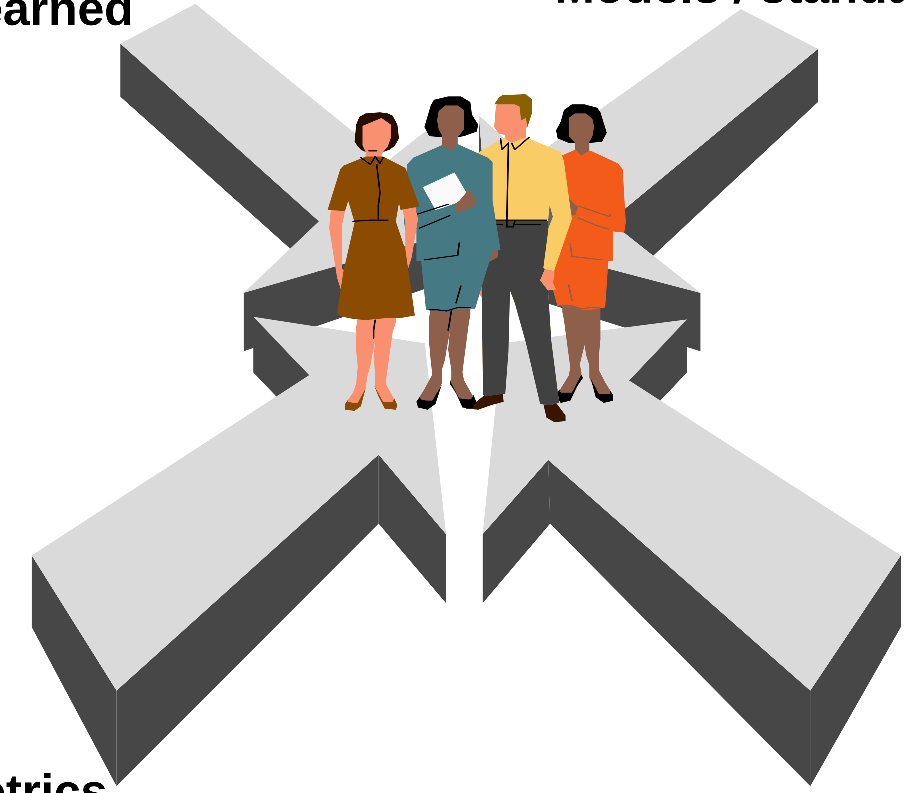
Politician / go-between

Relation of an EPG to QA

- Perceived as helping the practitioners and managers adopt new practices
- Not QA (Quality Assurance)



Focal Point for Improvement and Conveyer of Information

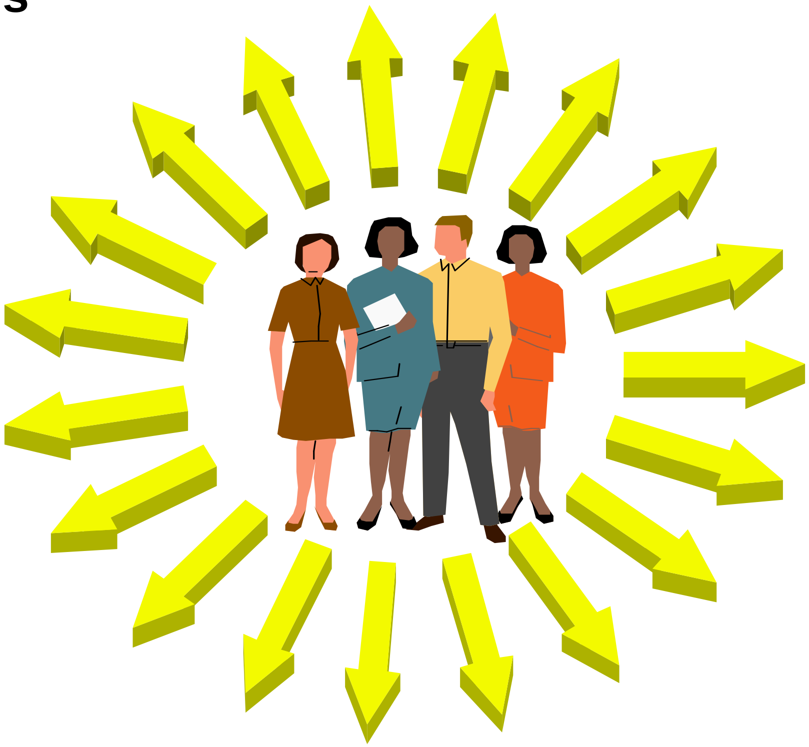


Lessons learned

Models / standards

Metrics

Study / reading



Deploys new concepts in a usable form

Staffing Your EPG

Skills

Must	Intense interest in process improvement
	Available full-time for EPG work (or dedicated part-time)
	Experience of and project management problems
	Able to work well with people
	Persuasive, resourceful (will try many different things)
	Attitude of helping others and good rapport with organization
	Communication skills (able to make all concepts practical)
Nice	Good coach and quality advocate
	Sensory acuity (able to adjust approach based on results)
Wish	Knowledge of process improvement, SEI, Deming, Crosby

Training

e.g., CMMI, process appraisals, consulting, engineering and management skills, metrics

Typical Things that an EPG Does

- Helps analyze expectations versus current practices
- Conducts engineering process assessments
- Helps write plans for improvement
- Helps implement the plans - facilitates improvement
- Maintains sponsorship
- Acts as an information source, sharing information across the organization
- Promotes and teaches new behaviors (process consultation and technology insertion focal point)
- Helps write processes and work aids
- Measures process improvement

Things that an EPG Does Not Do

- Reveal specific confidential information to management or others
- Take sole responsibility for the actual improvement
- Write processes or procedures in a vacuum
- Issue edicts or strong-arm changes

EPG Steering Committee

- Select key sponsors, champions and managers to ensure buy-in
- Explain issues, ideas, practices
- Review the status of improvement efforts
- Tell them what they need to do to help
- Tell them what resources you need
- Ask them to help resolve issues
- Obtain support for additional staffing
- Include targets with success stories
- Include related efforts such as quality teams and working groups (credit and praise)

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)