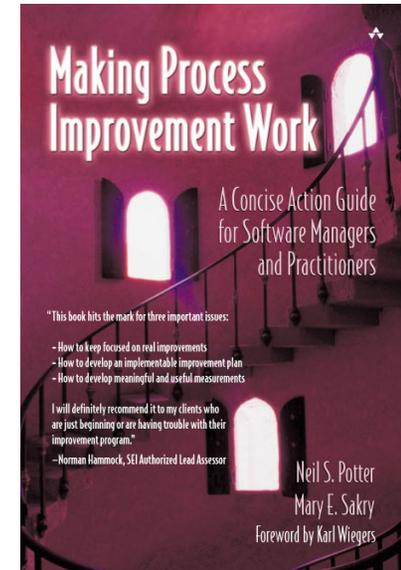
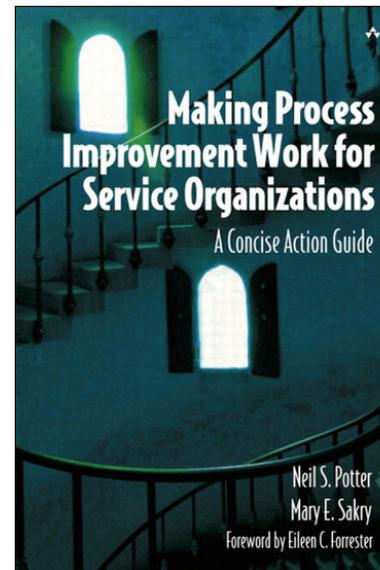


Making Process Improvement Work For Service and Development Organizations

Tying Improvement and CMMI Directly to What You Care About

Neil Potter
Mary Sakry

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



Agenda - 1

	<u>Page</u>
1. Introduction.....	3
2. Developing a Plan.....	7
– Scope the Improvement.....	7
– Exercise.....	18
– Develop an Action Plan.....	19
– Exercise.....	27
– Determine Risks and Plan to Mitigate.....	28
– Exercise.....	33

Agenda - 2

	<u>Page</u>
3. Implementing the Plan.....	34
– Sell Solutions Based on Needs.....	35
– Work with the Willing and Needy First.....	37
– Keep Focused on the Goals and Problems.....	48
– Align the Behaviors of Managers and Practitioners.....	51
– Avoiding a Documentation Glut.....	54
– Exercise	62
4. Checking Progress.....	62
– Are We Making Progress on the Goals?	65
– Are We Making Progress on Our Improvement Plan?	70
– Are We Making Progress on the Improvement Framework?.	72
– What Lessons Have We Learned So Far?	78
– Exercise.....	81

Agenda - 3

Page

5. Appendix.....	82
– Engineering Process Group (EPG).....	82
– References.....	87

Your Expectations

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

Introduction

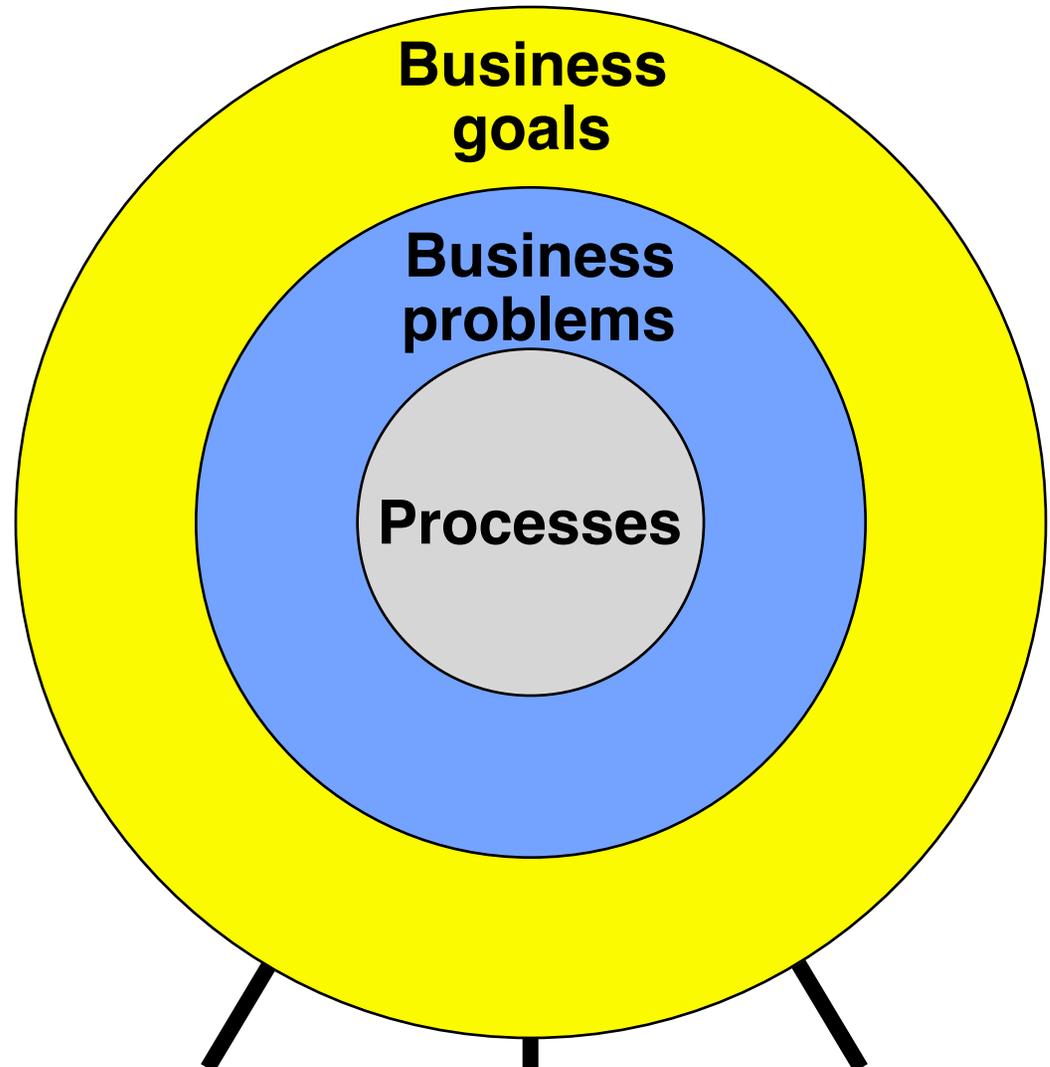
The “Classic” Approach to PI

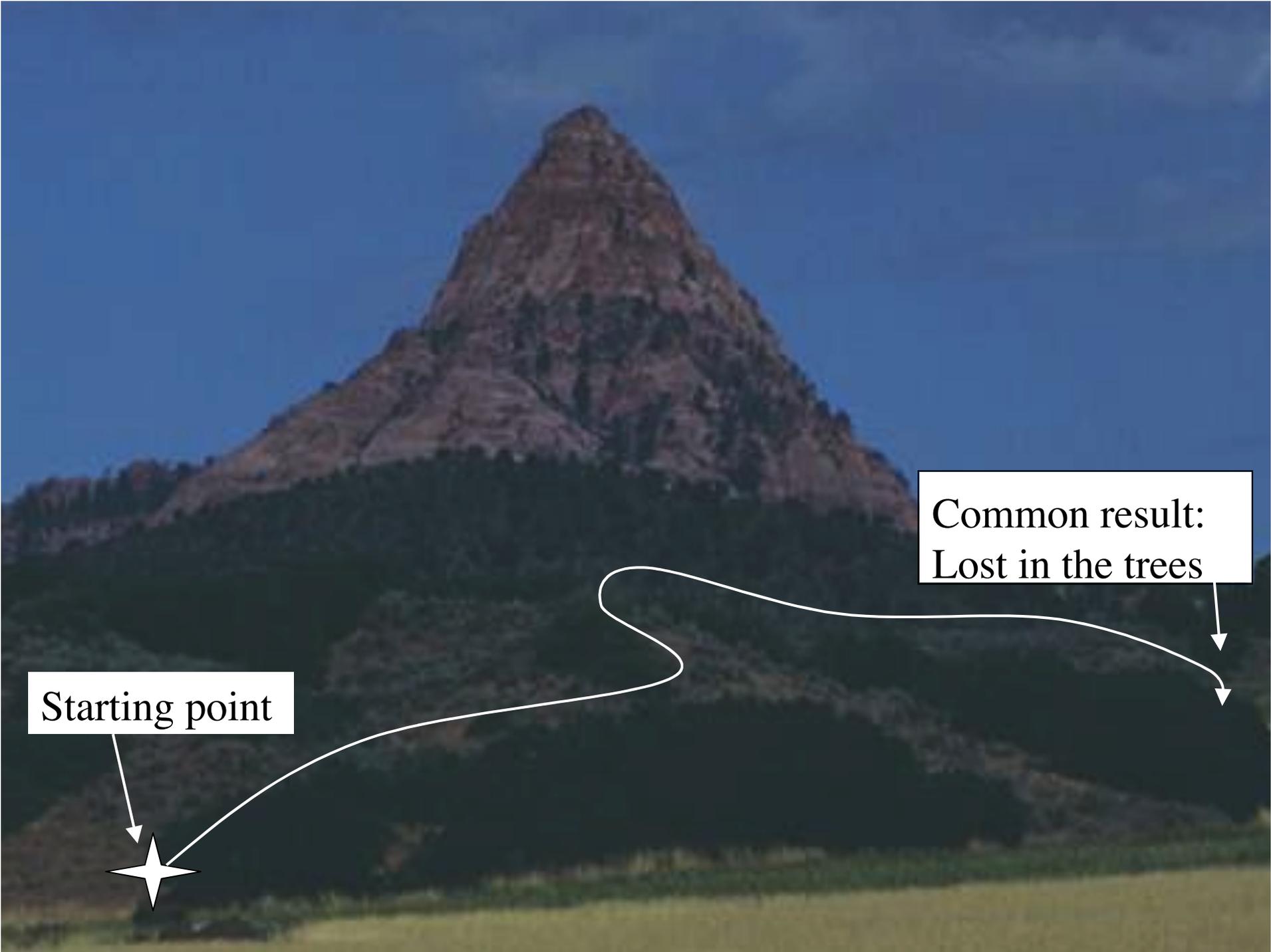
Process-centric improvement

- SEI CMMI
- ISO9001
- Bellcore

It can work!

- High risk of failure





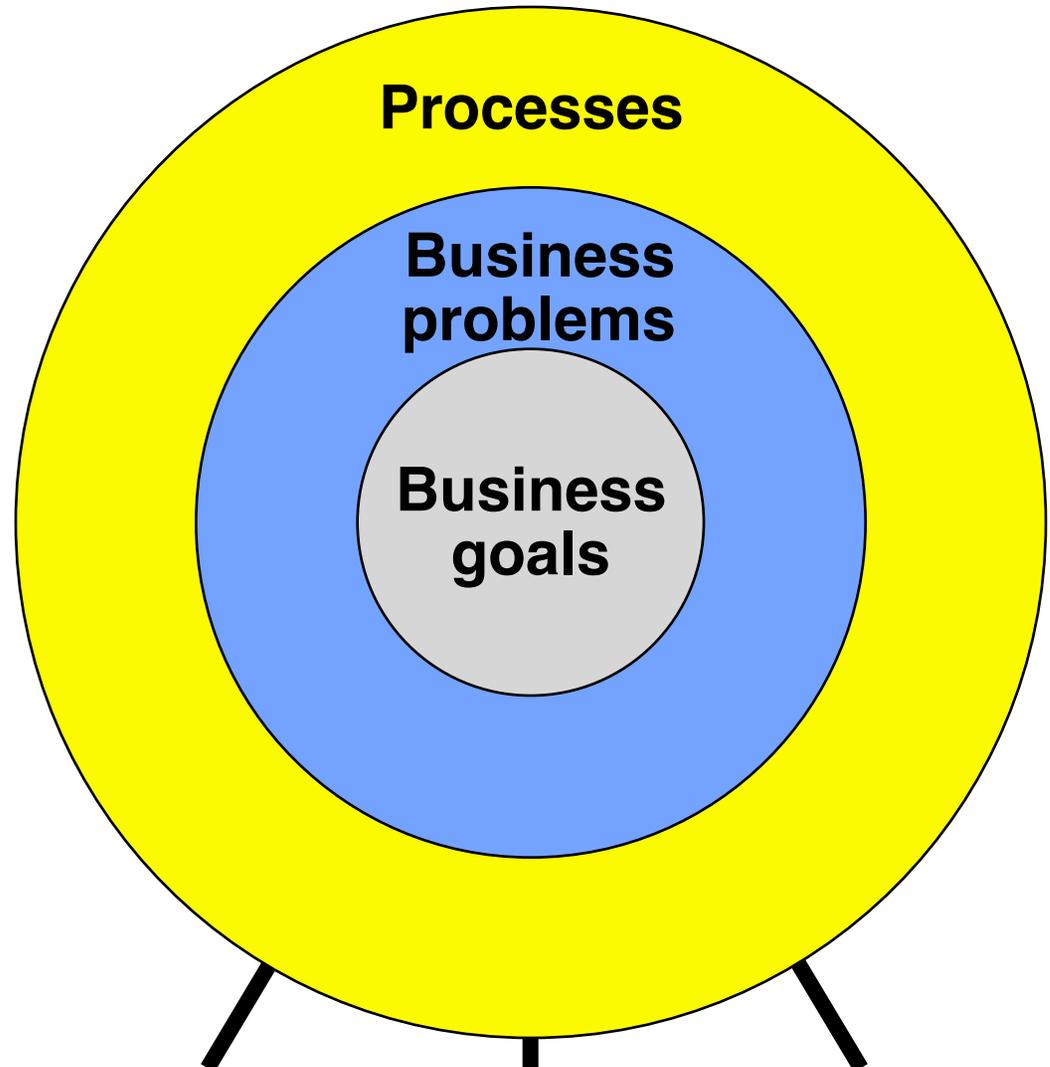
Starting point

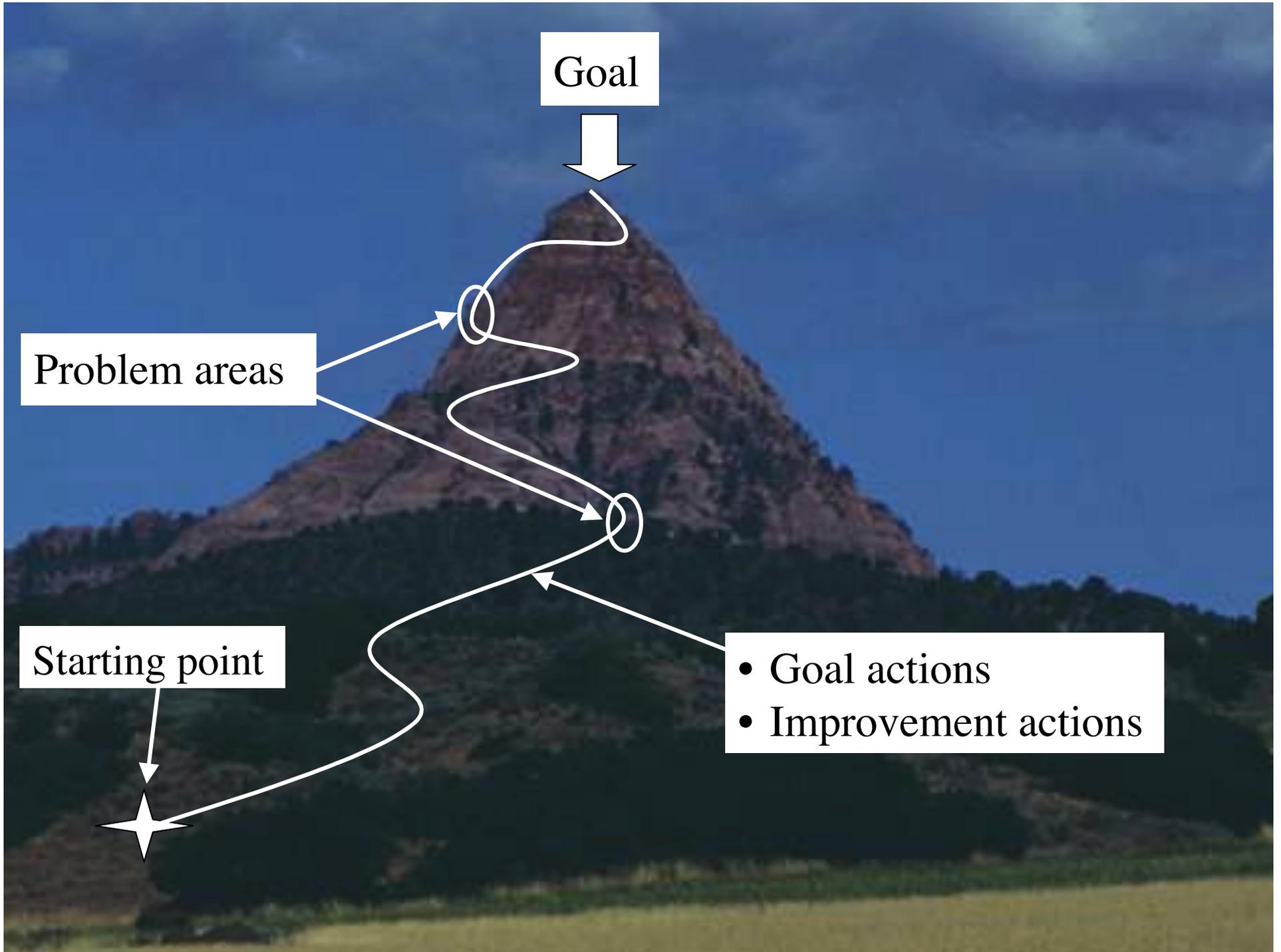
Common result:
Lost in the trees

A Solution

**Goal-problem-centric
improvement**

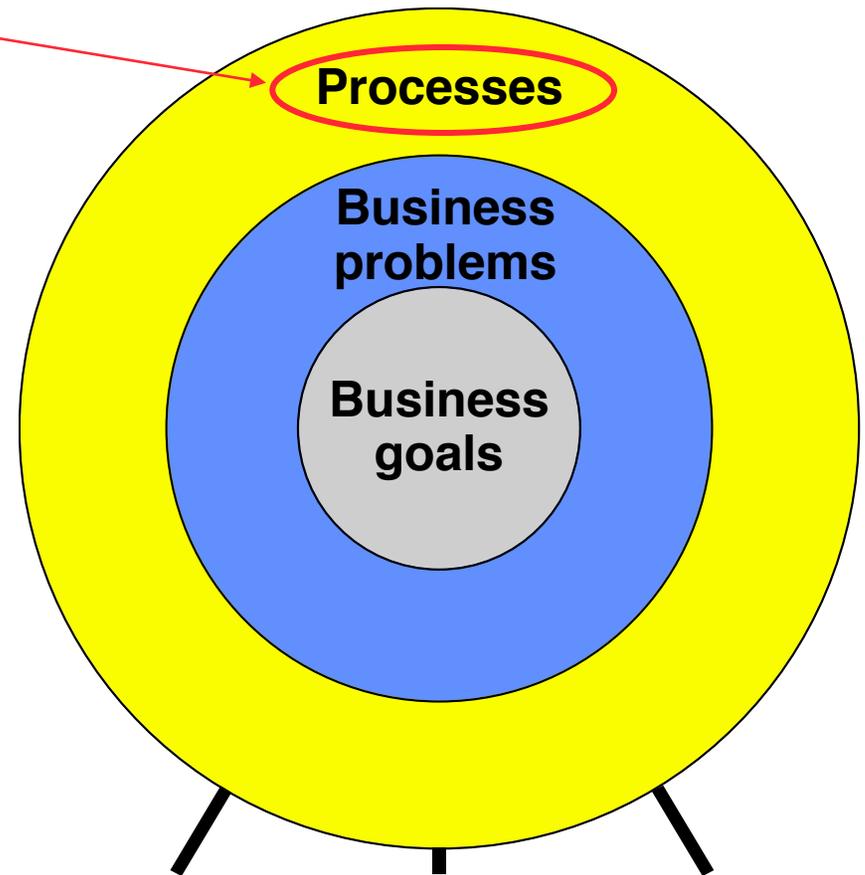
**Goals and problems
can be used to scope
and sequence the
improvement effort**





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 (development)

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

2. State the Major Goals and Problems - 1

Example Goals (services)

1. Achieve current event-planning performance targets
2. Provide existing services to new clients
3. Make staff externally focused on customers
4. Manage the time of the department to achieve goals

State the Major Goals and Problems - 2

Example Problems (development)

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.

State the Major Goals and Problems - 2

Example Problems (services)

1. The group's function is not well defined or agreed to among staff members.
2. There is no agreement on the group's role with other departments to enable staff members to stay focused on their activities. Instead, they are drawn into internal support issues.
3. Too much time is spent on fixing the problems of other groups.
4. There is no clear overall strategy and breakdown of major tasks that have to be accomplished during the year.
5. Risks are not assessed or managed. Crisis mode is the norm.
- ⋮
9. Weekly status meetings are not conducted regularly.
10. There is no system to record action items or track them to closure when they cannot be immediately addressed.
11. There is little analysis of previous event-planning activities (e.g., time and money expended, risks assessed and realized).
12. Customer deliverables are full of errors.

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 (development)
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 (development)
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 (development)
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.

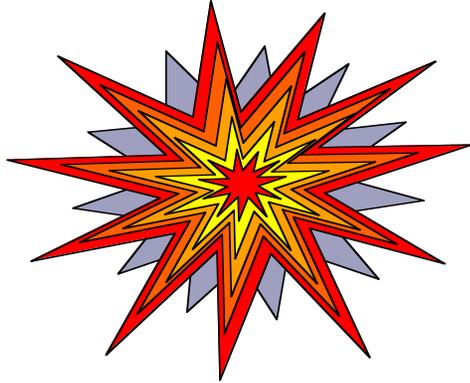
Group the Problems Related to Each Goal - 3

Goal	Problem	Problem Description (services)
4. Manage the time of the department to achieve goals.	Problem 3	Too much time is spent on fixing the problems of other groups.
	Problem 5	Risks are not assessed or managed. Crisis mode is the norm.
	Problem 6	The list of external stakeholder names (e.g., electricians, florists, equipment rental companies) is not well defined or known. This makes it hard to communicate status and issues.
	Problem 7	Documents are not consistently stored or shared among stakeholders.

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

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

Boom!



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.
 - Reduce the \$2 million currently spent on rework each year.
 - Reduce rework to 5 percent of project effort. Use that time to create new product / service Y.
 - Improve schedule prediction to \pm 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 **(Services)**

Original Goals	Goals Reworded for Clarity
1. Achieve current event-planning performance targets.	Achieve current event-planning commitments for fiscal year.
2. Provide existing services to new clients.	Provide existing services to new clients—specifically clean-energy companies.
3. Make staff externally focused on customers.	Spend 80 percent of each month on customer-related activities.
4. Manage the time of the department to achieve goals.	Improve department efficiency by 15 percent.

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.	9	5	1.8	1
5. Achieve customer rating of 9/10 on product evaluation form.	6	6	1	1
3. Reduce rework to less than 20 percent of total project effort.	7	5	1.4	2
6. Keep profits at 15 percent (and costs at the same as last year).	9	5	1.8	3
4. Improve the performance of our core software product. (Target to be defined.)	5	7	0.7	3

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

Using the Approach for a Single Team

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 staff 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. Don't 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

Development

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

Development

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

Actions for One of the Problems - 2

Services

Problem	What actions are needed to address the problems and achieve the goals?
<p>2. There is no agreement on the group's role with other departments to enable staff members to stay focused on their activities. Instead, they are drawn into internal support issues.</p>	<p>Determine stakeholders from other departments who need to agree with, and support, the new roles definition.</p>
	<p>Establish meeting events with stakeholders to present:</p> <ul style="list-style-type: none"> (a) the current problem, (b) the priorities of the group, and (c) recommended changes to the group's charter.
	<p>Establish a one- to two-page service agreement with stakeholders.</p>

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

Development

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

Development

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

Services

Reworded for clarity
↓

Problem	Which elements will help the problems and goals listed?
<p>2. There is no agreement on the group's role with other departments to enable staff members to stay focused on their activities. Instead, they are drawn into internal support issues.</p>	<p>Establish and maintain the service agreement (containing specific service-level commitments). (SD sp1.2)</p>
	<p>Review the activities, status, and results of the service definition activities with higher-level management and resolve issues. (SD gp2.10)</p>
	<p>Manage changes to service requirements as they evolve. (REQM sp1.3)</p>
	<p>Place selected work products (e.g., agreements and service definitions) under an appropriate level of document control. (SD gp2.6)</p>

SD = Service Delivery

Progress on Chosen Framework -1

95%
map
to
Level
2

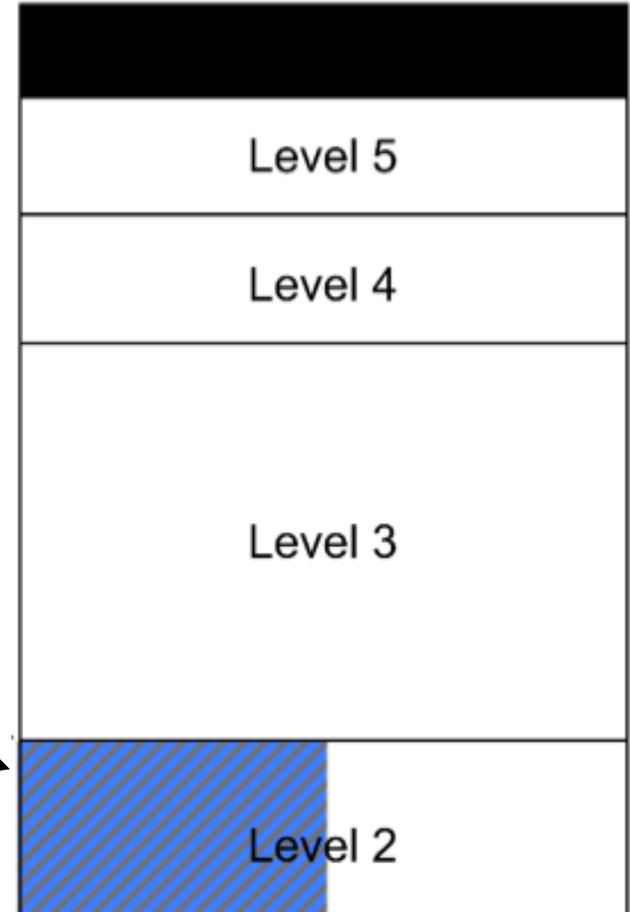
Example Goals

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

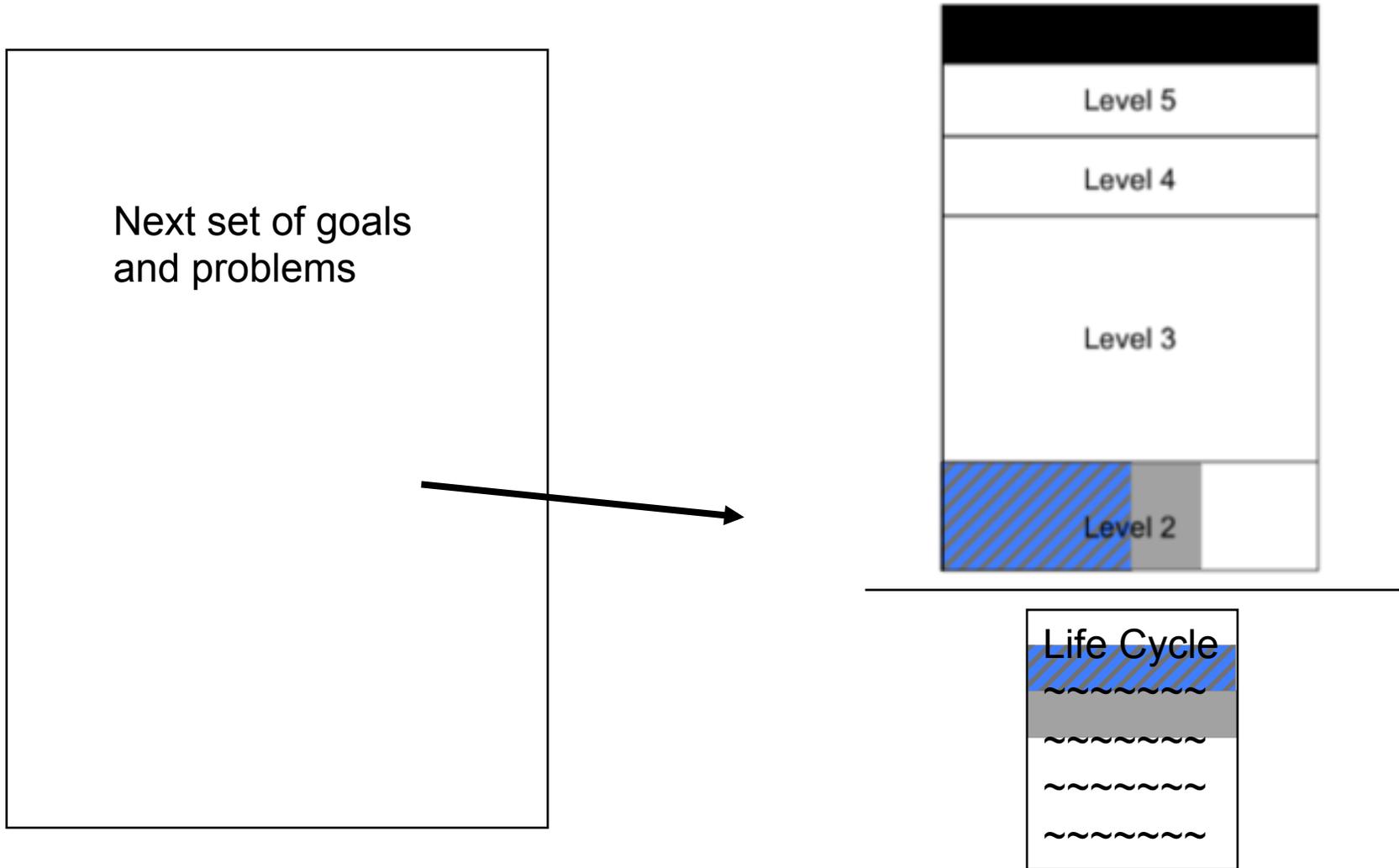
Example Problems

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 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.

Initial
goals
and
problems
address
43% of
Level 2

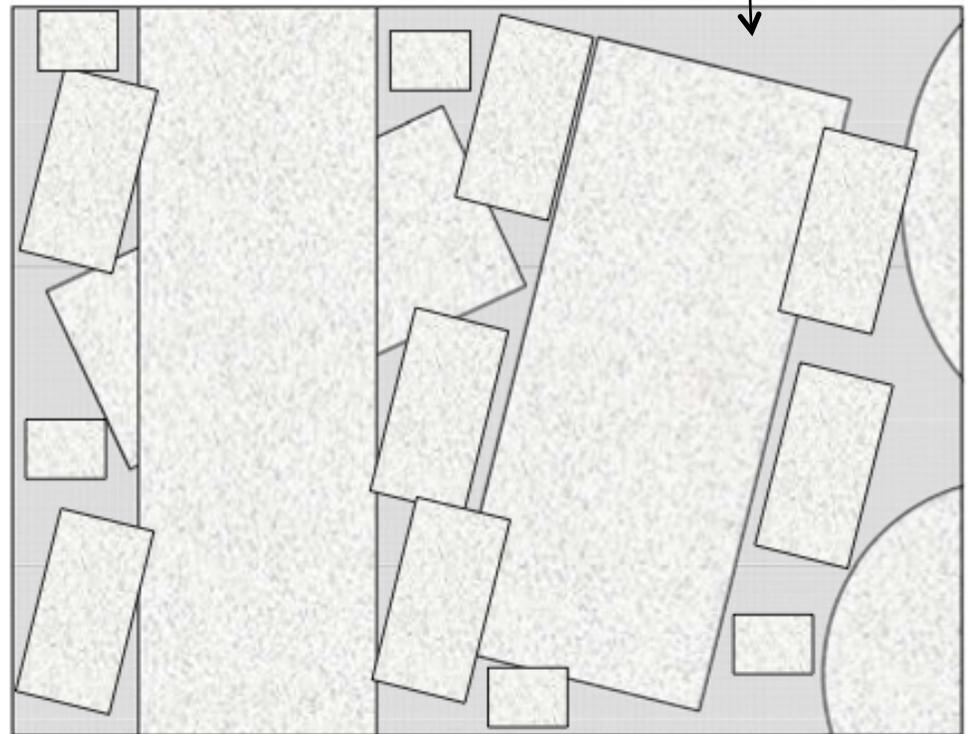


Progress on Chosen Framework -2



What to Do With the Remaining Elements?

- **Put each to good use**
 - What problem could it solve?
- **Merge with another practice?**
 - 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.html

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*
		<i>Check progress and take corrective action.</i>	-
Step 3: Add placeholder for checking progress and taking 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

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*
		<i>Check progress and take corrective action.</i>	-
		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
Poor-quality service delivery in our retail stores (service example)	Define detailed procedures instructing staff how to behave. Publish as a series of field service manuals.

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	<ul style="list-style-type: none"> • Inspect (peer review) all critical documents and code.
	<ul style="list-style-type: none"> • Improve estimation of test time needed.
	<ul style="list-style-type: none"> • Train test engineers in test skills.
	<ul style="list-style-type: none"> • Send test engineers to a customer site to understand how the customer uses the product. Factor this knowledge into the test strategy.
Poor-quality service delivery in our retail stores (service example)	<ul style="list-style-type: none"> • Have management visit stores to experience being a customer, and determine specific issues. • Analyze customer service complaints. • Train staff in expected service delivery skills (technical knowledge and delivery style).

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*
		<i>Check progress and take corrective action.</i>	-
		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

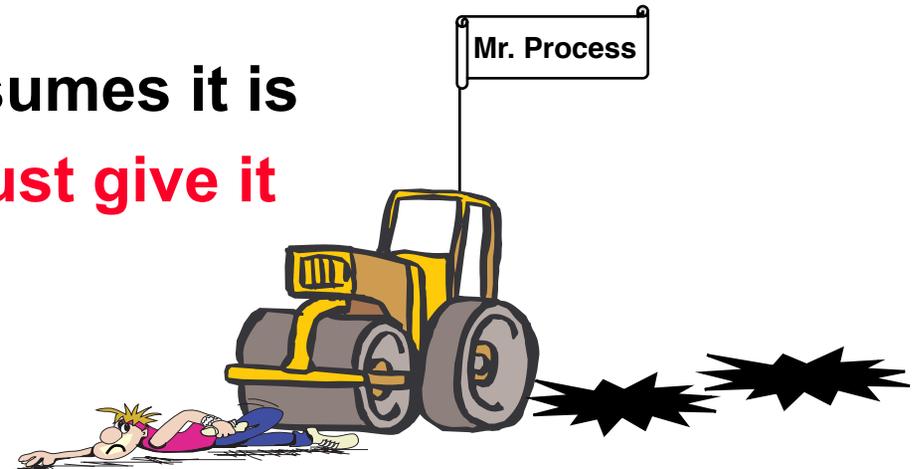
Implementing the Plan

“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**



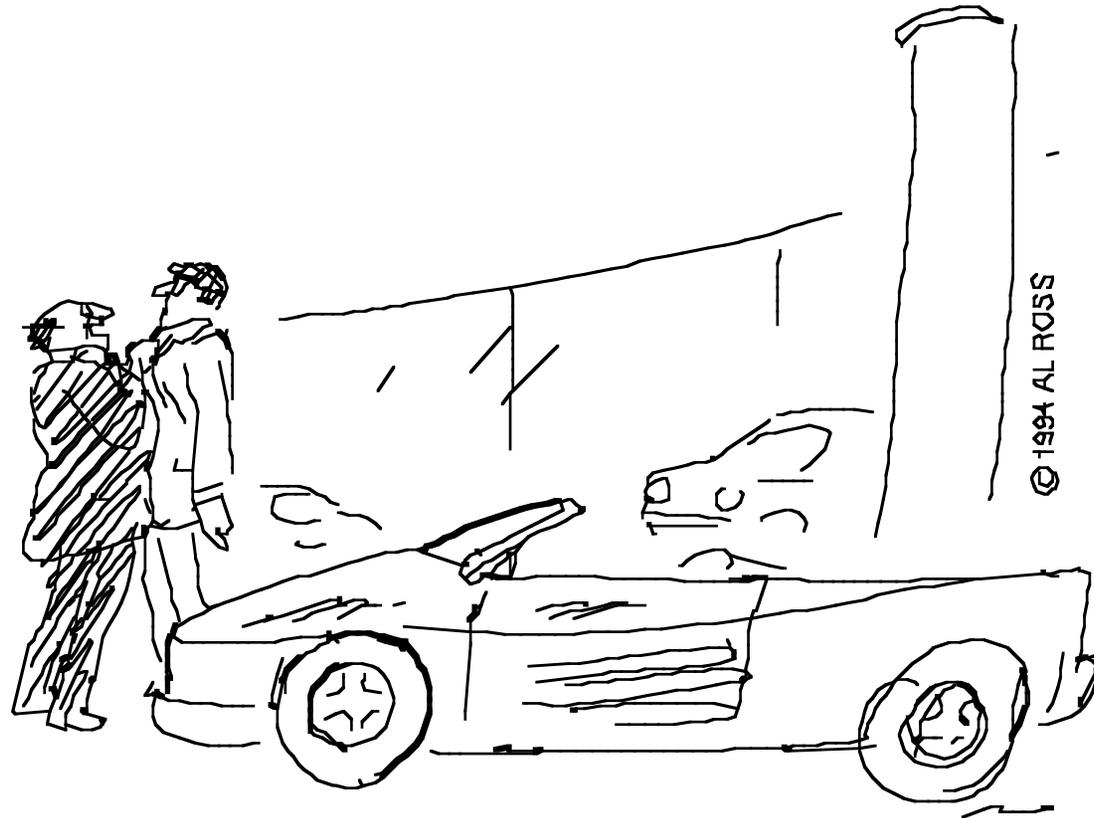
Implementing the Plan

- 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*



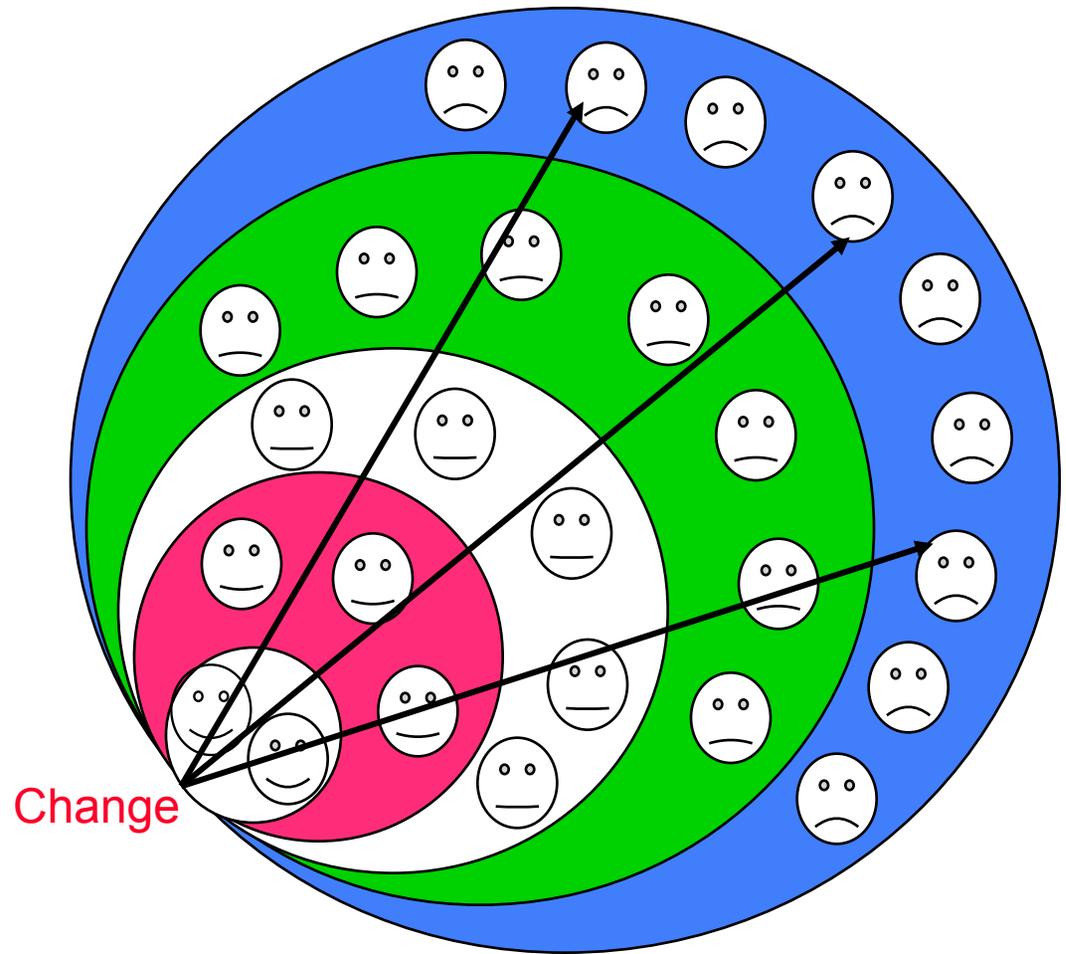
- **Forget** what you are selling
- **Understand** what the customer wants in his/her terms
 - Problems and goals
- Determine the **match** with what you have and what the customer wants
- **Solve** the customer's problem
 - may be a standard or customized solution

Implementing the Plan

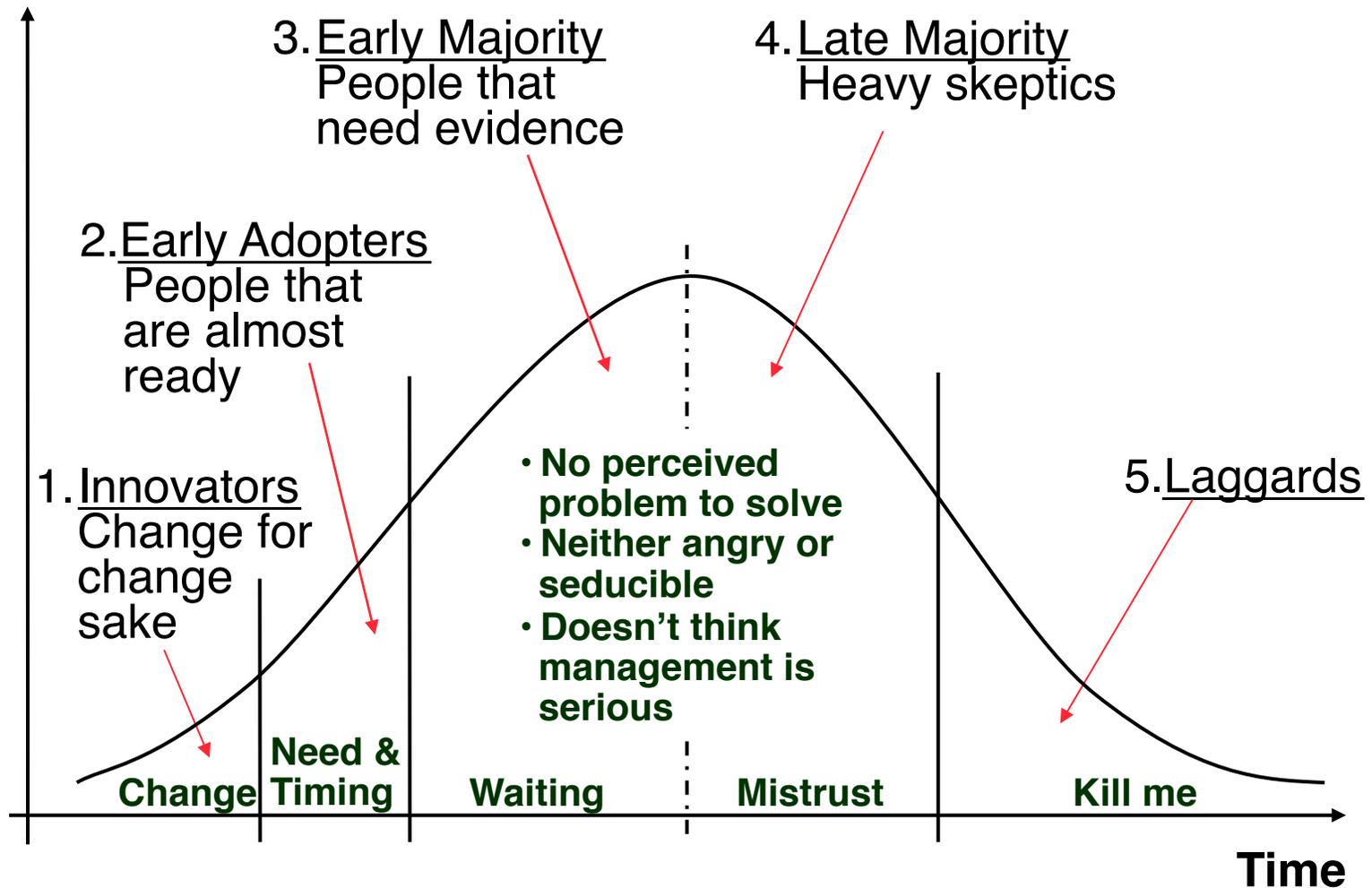
- Sell Solutions Based on Needs
- Work with the Willing and Needy First
- Keep Focused on the Goals and Problems
- Align the Behaviors of Managers and Practitioners
- Avoiding a Documentation Glut

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?



How are the Groups Determined?

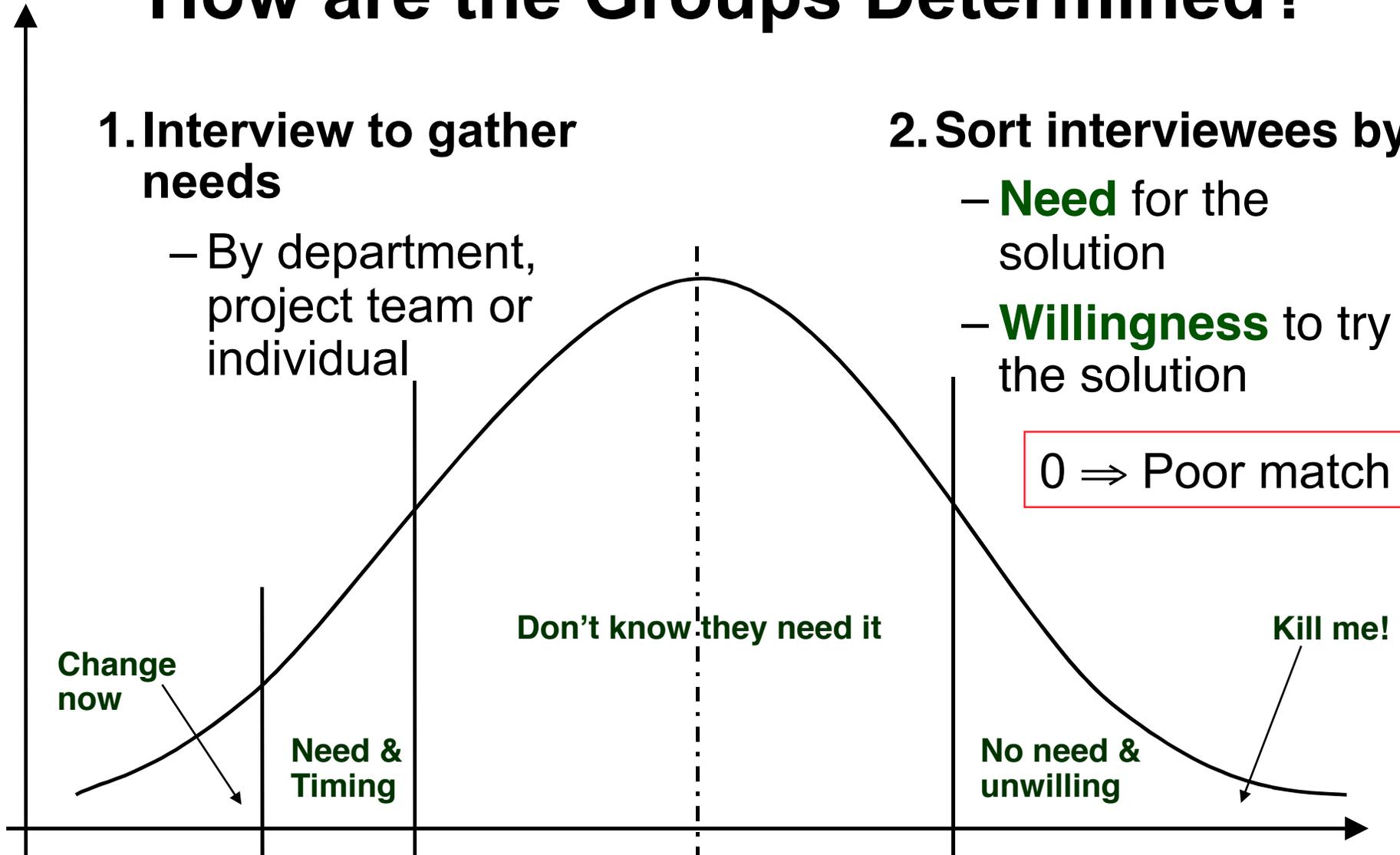
1. Interview to gather needs

- By department, project team or individual

2. Sort interviewees by

- **Need** for the solution
- **Willingness** to try the solution

0 ⇒ Poor match



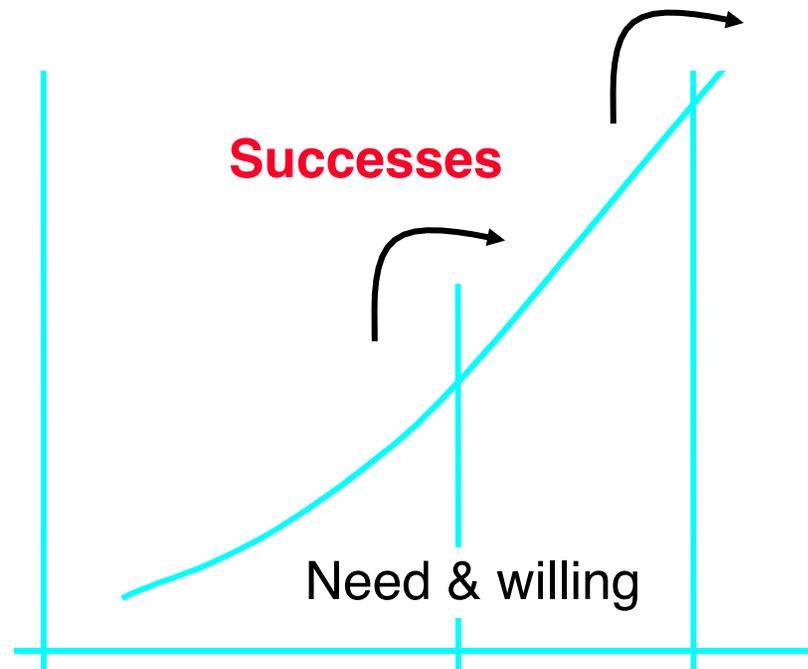
Three Uses of the Adoption Curve

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

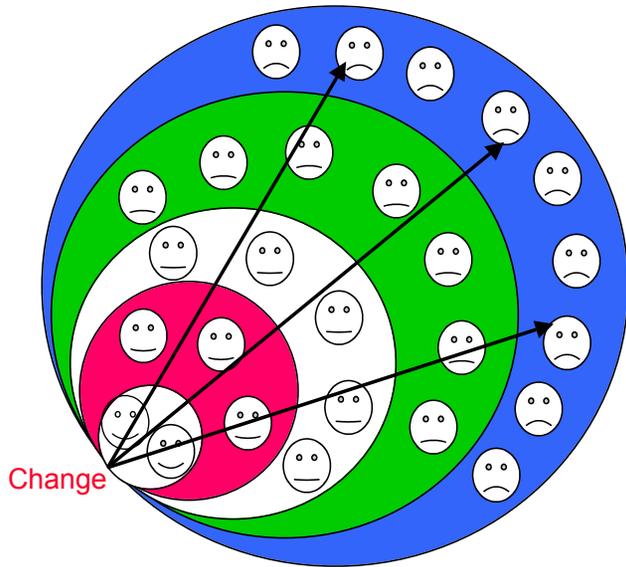


Successes: Getting the Word Out

- How to reach your audience



Use 2: Reduce the Risk of Failure

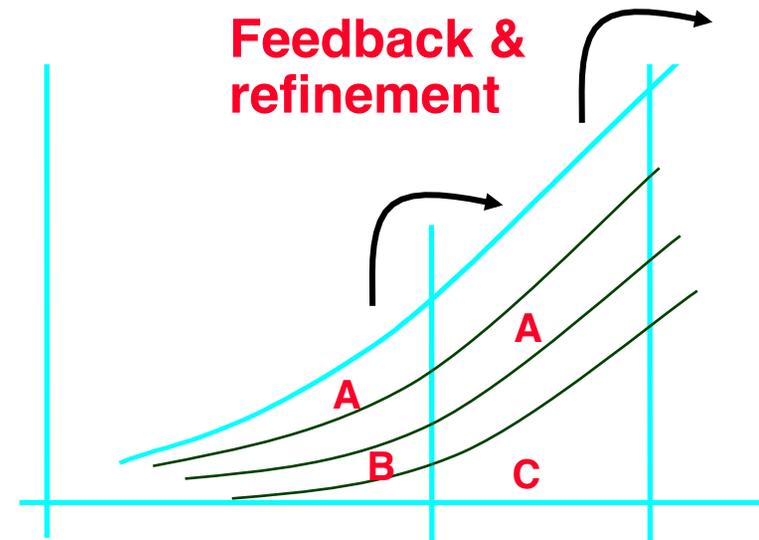


Innovators and early adopters can provide specific requirements for and feedback on early versions of the solution

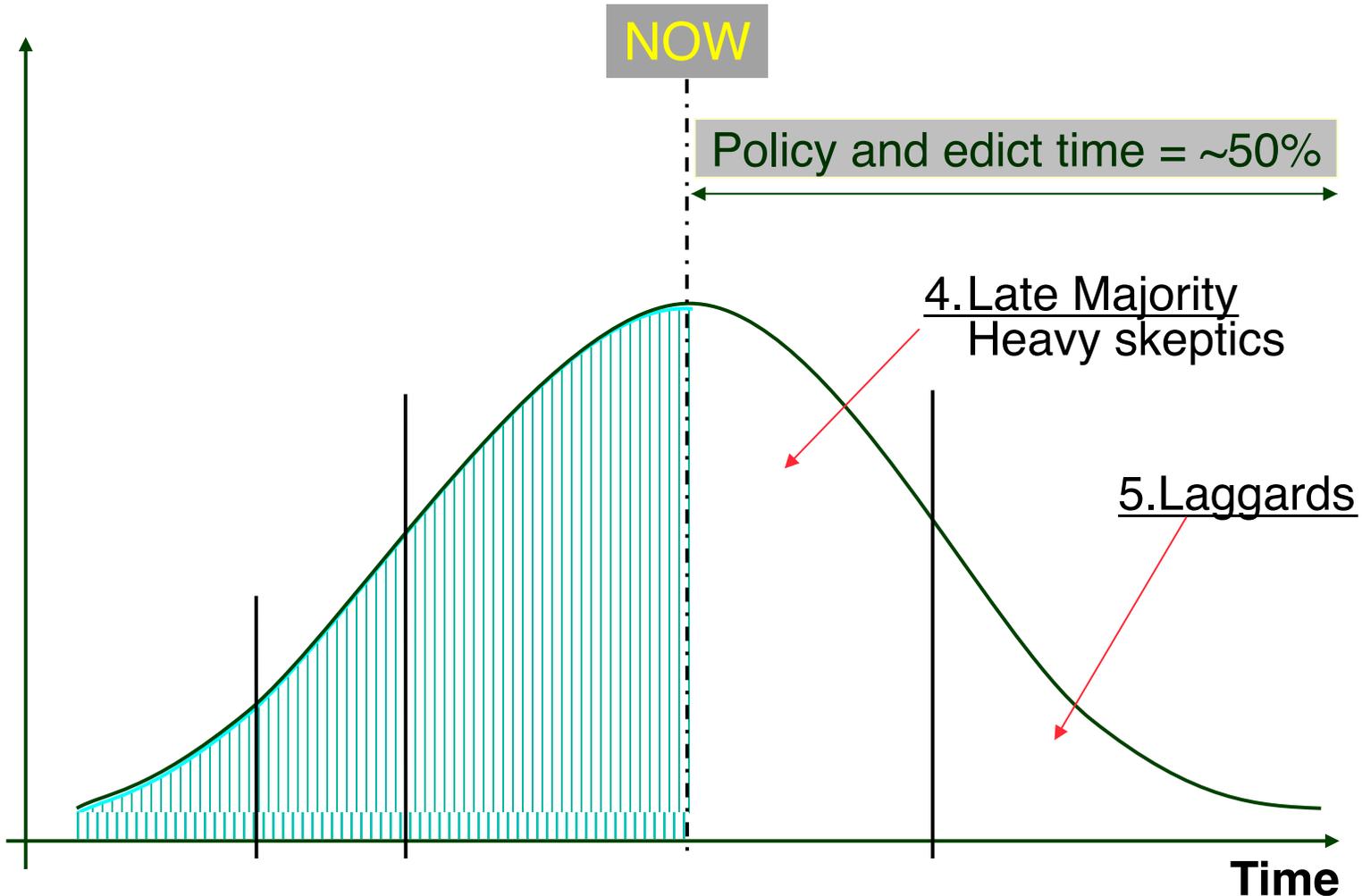
For example:

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

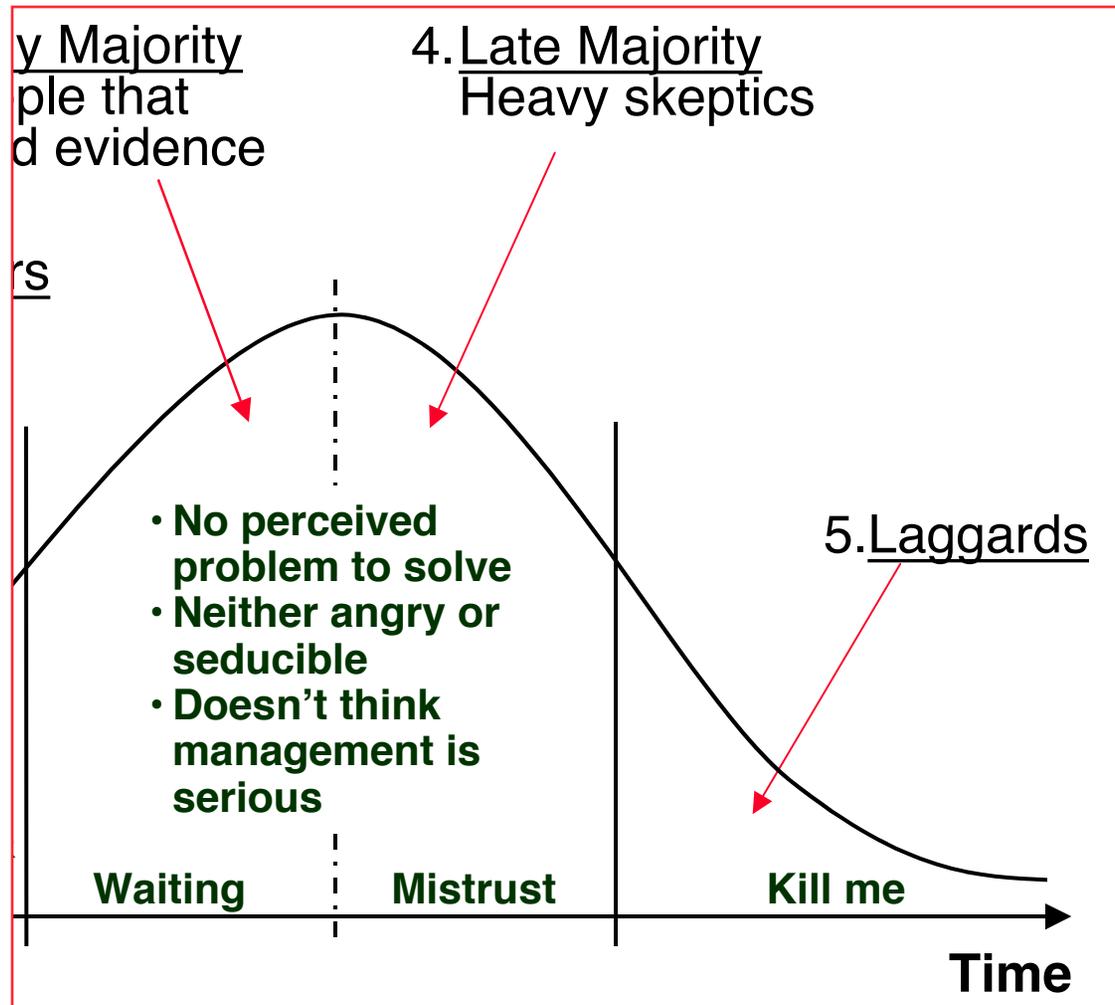
- **Policy states:**
 - When and where a practice should be used.
- **In the beginning:**
 - You might not have any idea!

Wait until you get some experience and feedback

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

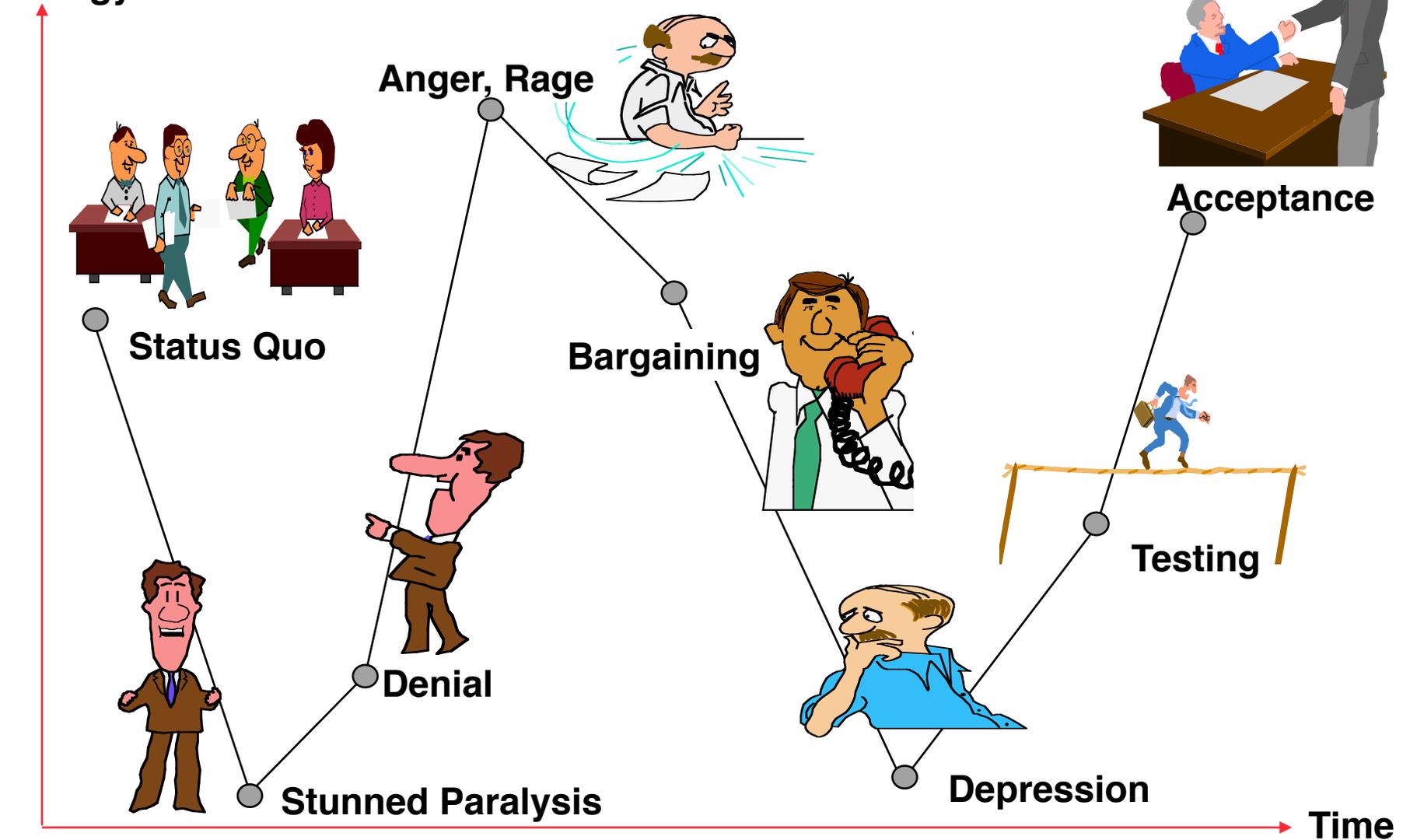
Wait until you get some experience and ownership

Resistance



Negative Reaction to Change

Energy



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 - 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 values**
 - What is most important to you?
 - What is most important to you about planning?
- **Understand beliefs**
 - 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

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

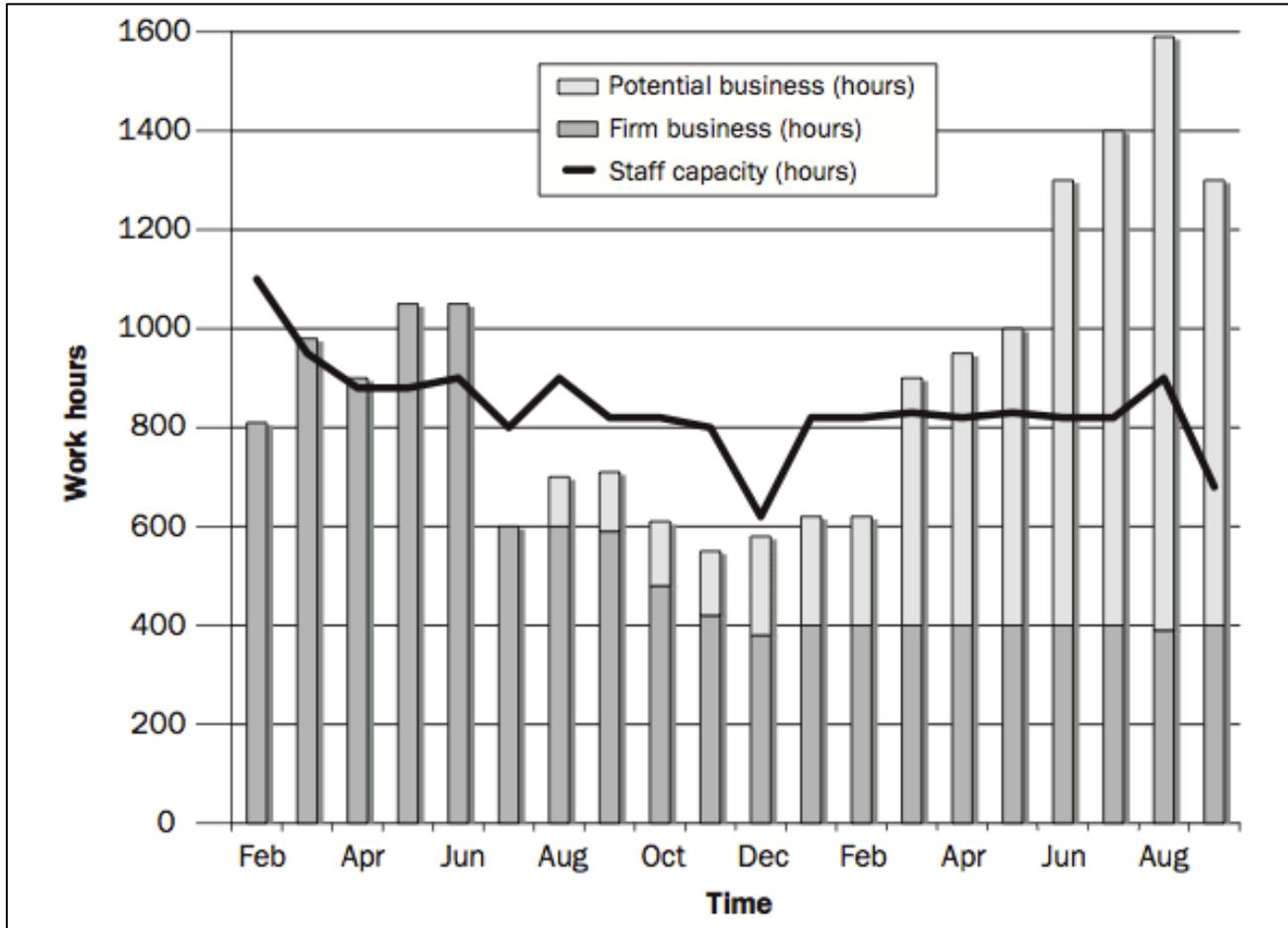
Services

Goal	Questions	Metrics
Achieve current event-planning commitments for fiscal year.	Are we spending the planned number of hours on each work assignment to complete it on time? Are we making or losing money?	Planned versus actual effort for each work assignment Annual net profit
Provide existing services to new clients—specifically clean-energy companies.	What is our existing market volume? What percentage of trade shows are we marketing at?	Revenue from clean energy Percent of clean-energy trade shows presented Number of new clients
Spend 80 percent of each month on customer-related activities.	What is our average contact with customers now? Does it vary by month?	Percentage of hours per month spent on customer-related activities
Improve department efficiency by 15 percent.	How much time do we spend on rework now? Are we improving? What are the significant time distractions with other departments?	Percentage of effort spent on fixing mistakes Hours per week spent on significant time distractions with other departments

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?

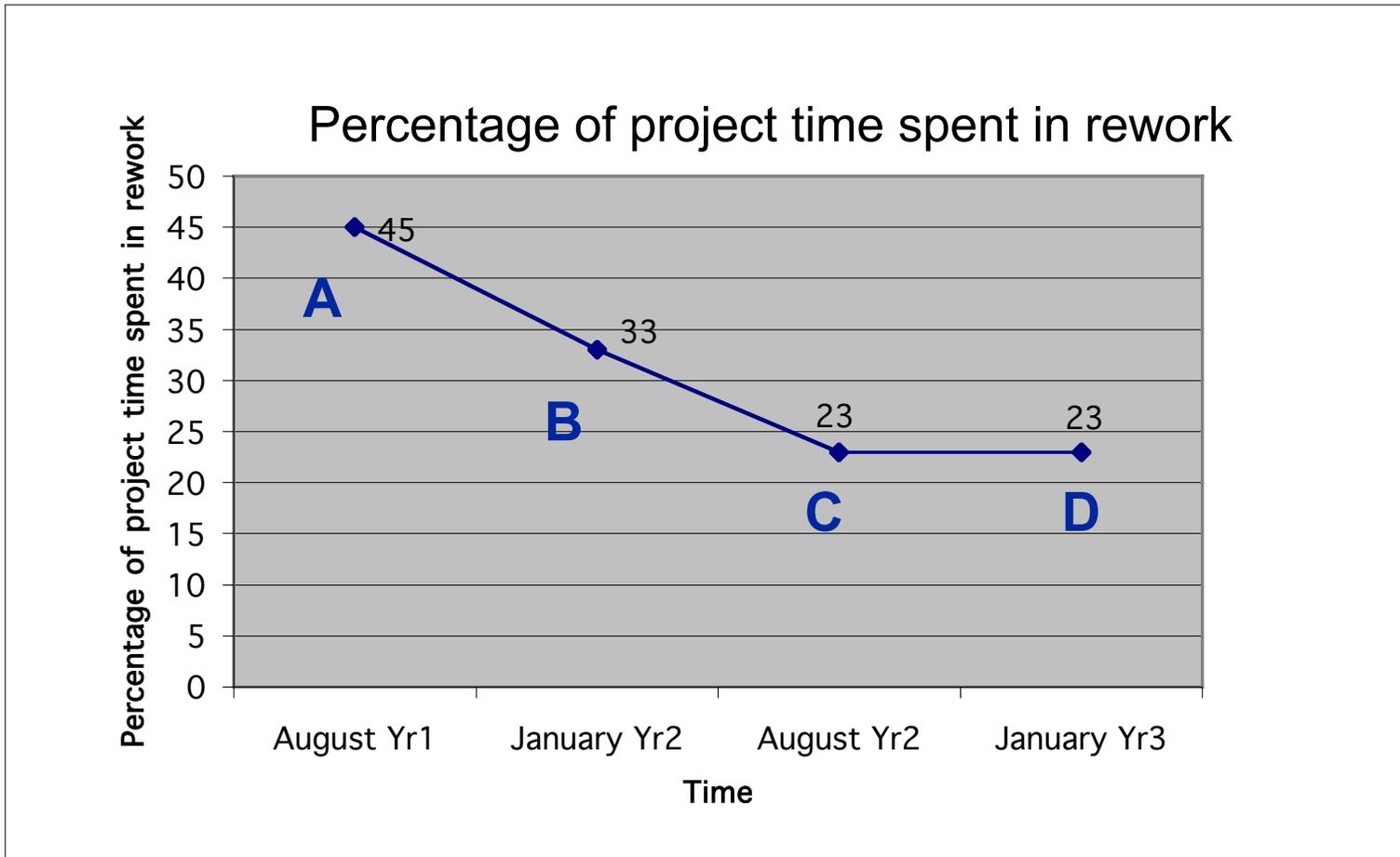
Goal: Meet Commitments



Observations and Corrective Actions

- **Managing peak times:**
 - Use interns and financial analysts from **other departments**
 - **Cross-training** all staff members to handle new work
 - Using **contractors**
 - Using **administrators** to share the workload for tasks that don't need a financial background
 - Creating **relationships with universities** so that candidate interns are available
- **Managing downturns:**
 - **Lending resources** to other groups
 - Using up backlogs of **vacation time**
 - Not filling **open positions**

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

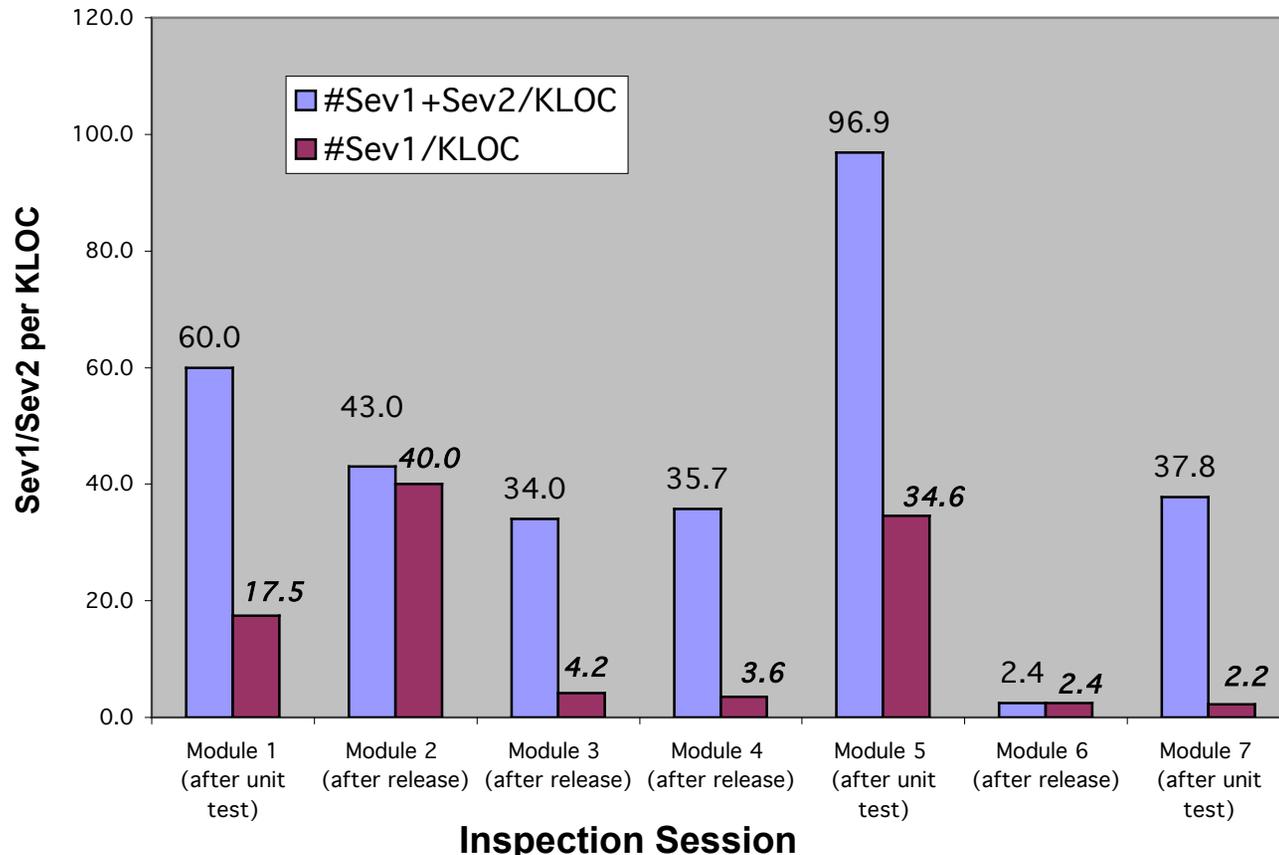


Observations and Corrective Actions

- **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



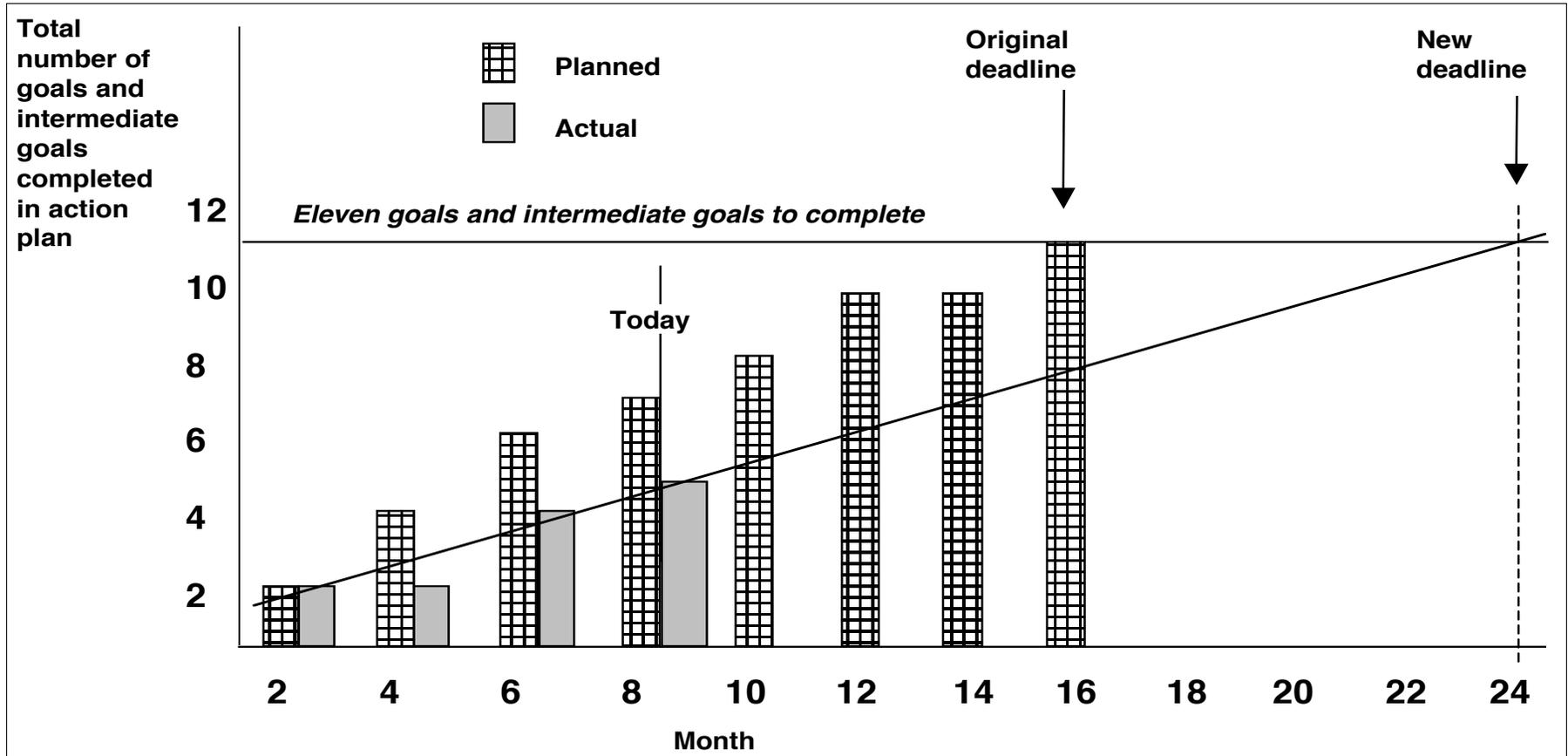
Observations and Corrective Actions

- **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

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

Checking Progress

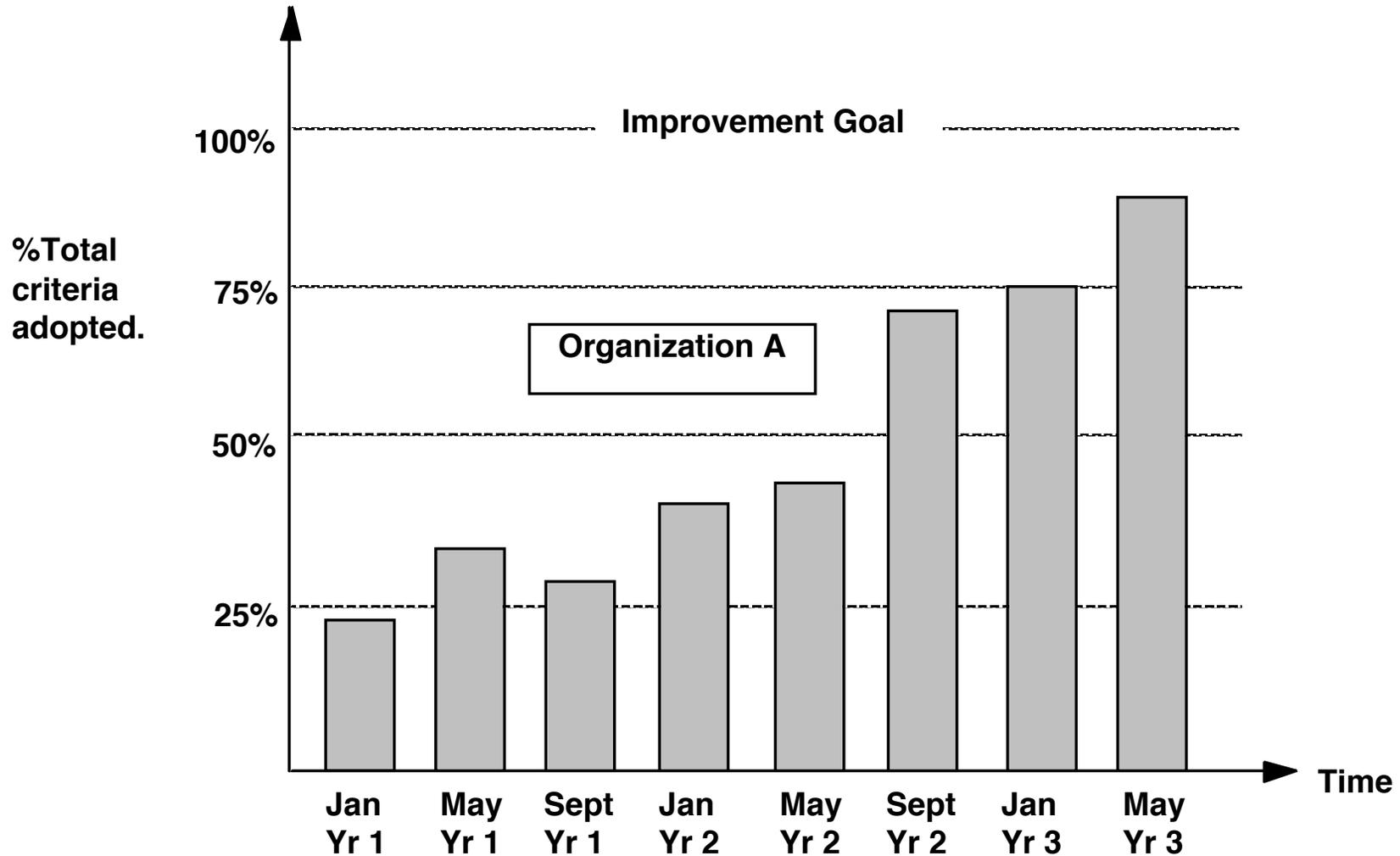
- Are We Making Progress on the Goals?
- Are We Making Progress on Our Improvement Plan?
- Are We Making Progress on the Improvement Framework?
- What Lessons Have We Learned So Far?

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* ✓
		<i>Check progress and take corrective action.</i>	-
		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

Example Mini-assessment Data - 2



Lessons Learned - Strengths

Lesson	Where to Use Lesson
<p>Decentralizing the action plan gives each project team ownership over its plan.</p> <p>Corrective action (CA) = Continue having three separate action plans, one for each of the three product lines.</p>	<p>Planning</p>
<p>Don't preach when an example can say everything for you.</p> <p>CA = Have one project each month conduct a one-hour briefing describing the use and benefits of a new technique.</p>	<p>Implementing</p>
<p>Guide people in applying each new technique to their work. People have so much going on that they do not know where to start.</p> <p>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.</p>	<p>Implementing</p>

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. Hammer, M., and J. Champy. *Reengineering the Corporation: A Manifesto for Business Revolution*. Collins Business Essentials, 2003.
4. Humphrey, W., "Software Quality Assurance." In: *Managing the Software Process*. Reading, MA: Addison-Wesley, 1989:137–153.
5. Humphrey, W., *A Discipline for Software Engineering*. Reading, MA: Addison-Wesley, 1995.
6. Moore, G., *Crossing the Chasm*. New York: Harper-Business, 1991.
7. Potter, N., Sakry, M., "Making Process Improvement Work - A Concise Action Guide for Software Managers and Practitioners," Addison-Wesley, 2002. (<http://www.processgroup.com/book.html>)
8. Potter, N., Sakry, M., "Making Process Improvement Work for Service Organizations - A Concise Action Guide," Addison-Wesley, 2012. (<http://www.processgroup.com/book.html>)
9. Software Engineering Institute, *CMMI for Services, Version 1.3*, CMU/SEI-2010-TR-034, 2010.
10. CMMI: <http://www.sei.cmu.edu/cmmi/solutions/>
11. ROI information: <http://www.processgroup.com/resources.htm> (see ROI Data)