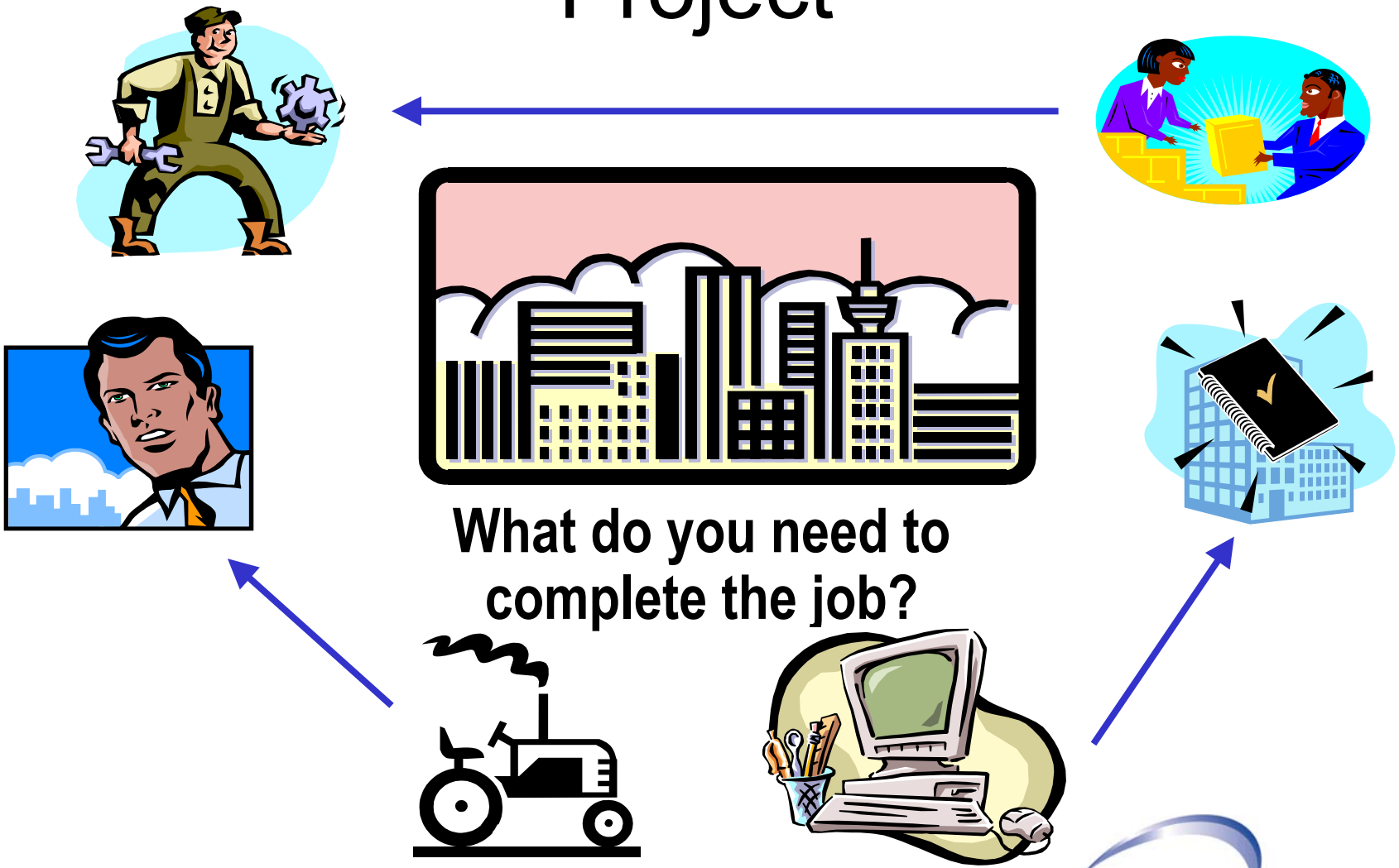
People, Process, and Tools: Keys to Successful Software Projects

Gary Pollice, Rational Software

HP WORLD 2002 Conference & Expo

Think About Any Significant Project



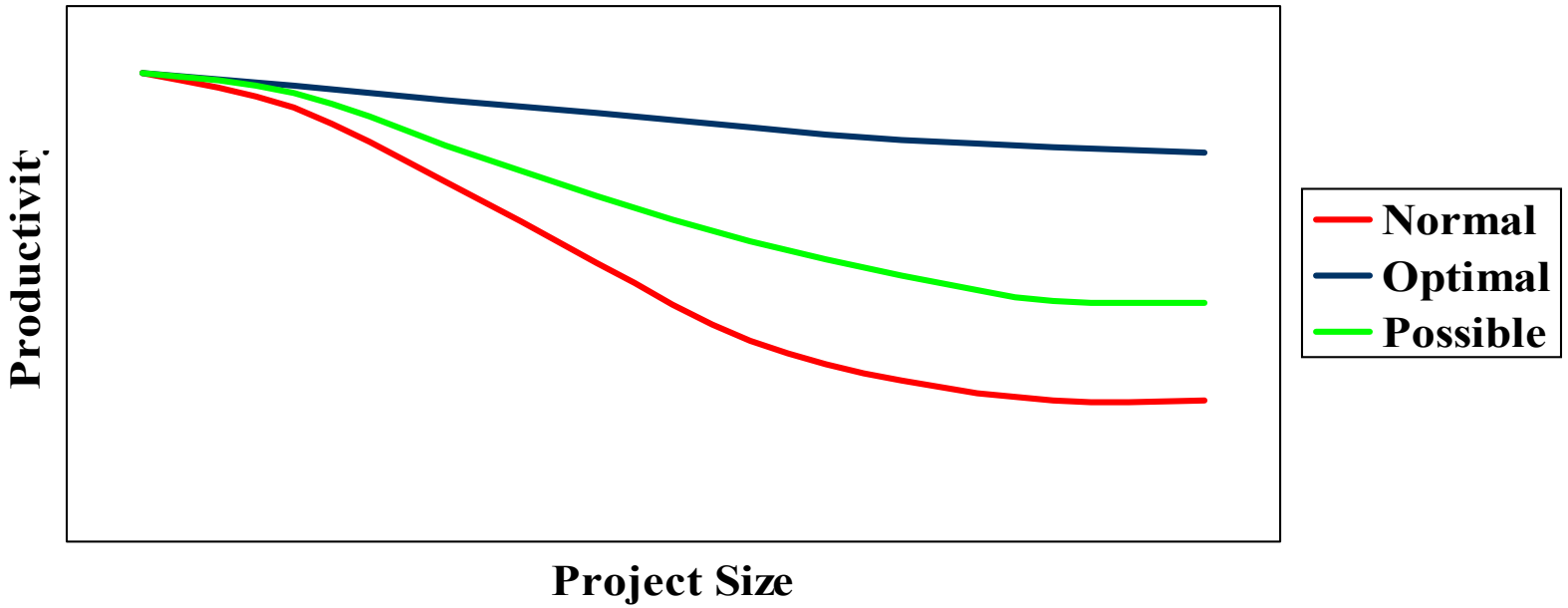
Why do we build software?

To satisfy our customers

People

- The most important asset on any project
- People can make or break a project
- Process and tools must support people
- People need:
  - Knowledge
  - Responsibilities
  - Guidance
  - Freedom
  - Respect

Individual Productivity vs. Project Size



Factors That Reduce Productivity As Project Size Increases

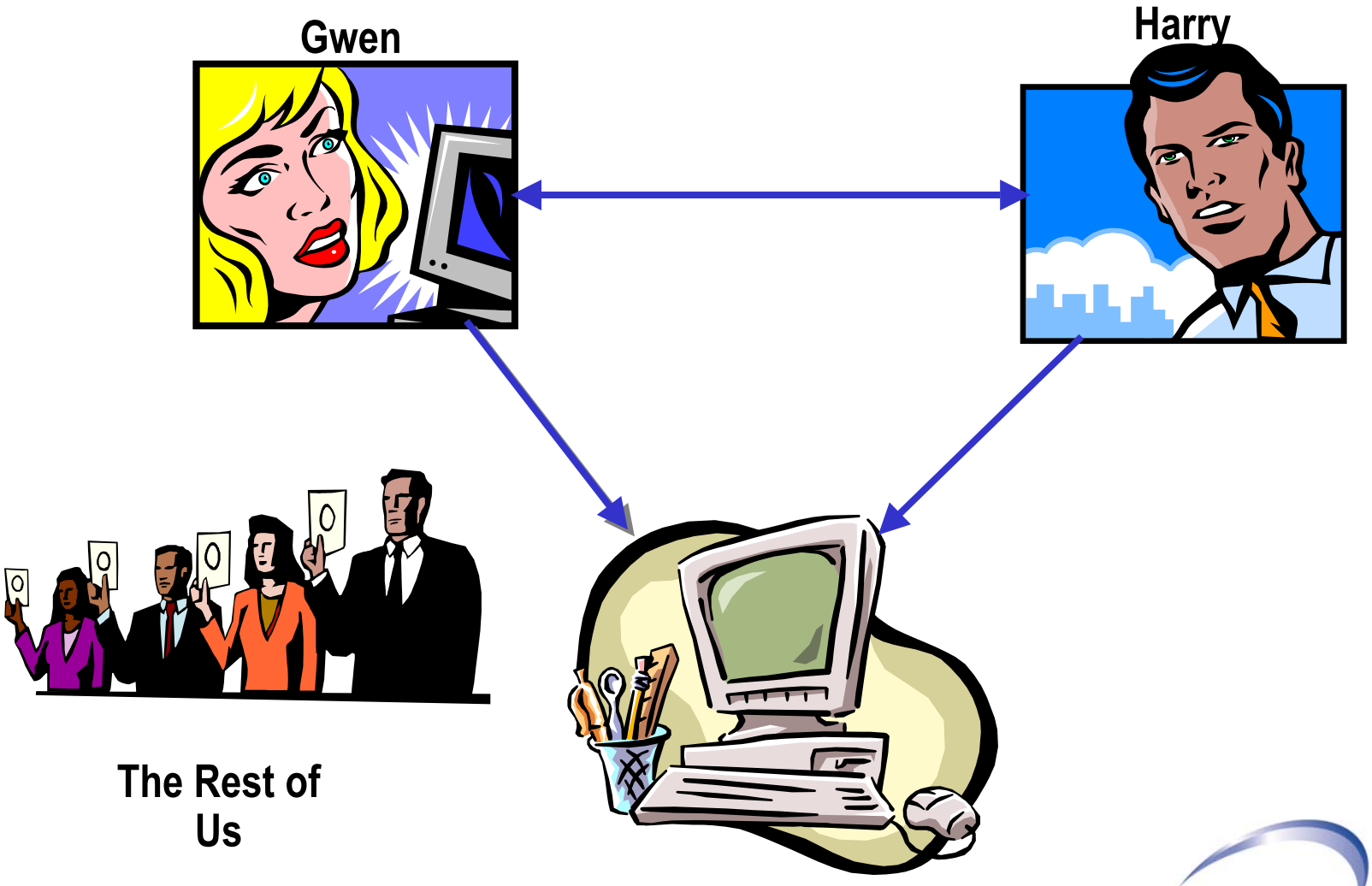
- Increased communication paths
  - Project members
  - Stakeholders
- Increased system complexity
- Novelty in the application and technology
- The whole system cannot be understood in detail by one person

Factors That Help Maintain Productivity As The Project Grows

- Effective process (right-sized for the project)
- Tools that support the team
- Effective training
- Good planning and project structuring
- Good people management

As project size increases, place more emphasis on team productivity instead of individual productivity.

A Tale Of Two Programmers



Team Makeup

- Staff the project with people that have complementary skills
  - Technical skills
  - People skills
- Staff the project with people that have different levels of experience and expertise
  - Expert, Journeyman, Apprentice
  - Domain expertise as well as technical expertise

Team Makeup

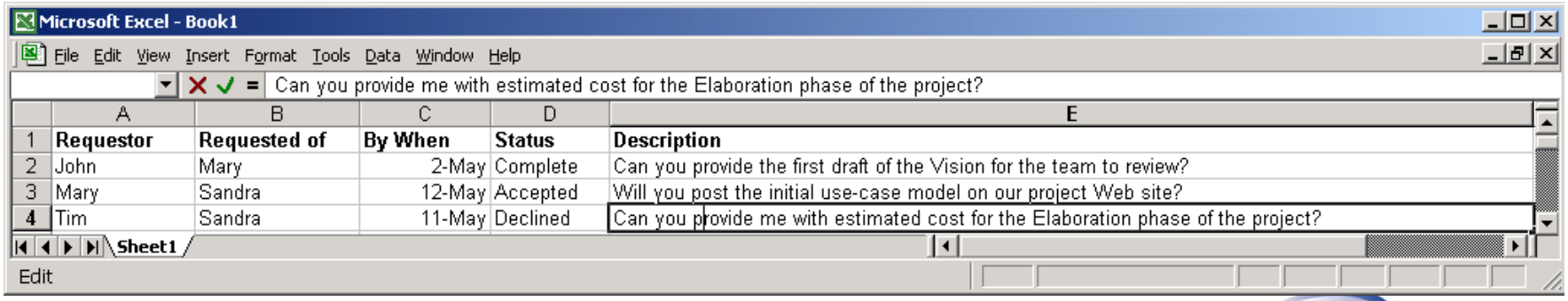
- Limit administrative hierarchy
  - Too many managers spoil the project
- Make the team as all-encompassing as possible
  - Customers
  - Business managers
  - Other stakeholders

Team Activation

- Provide a learning environment for everyone
  - Does not have to be technical
  - Make learning a goal for everyone
- Share responsibility
  - Everyone should be responsible for something
- Allow disagreement
  - Define consensus
  - Have a well-defined resolution strategy

Team Activation

- Make requests, not demands
  - If you can't say no, it's not a request
  - Make well-formed requests (who, what, when, and maybe where)
  - Do not use requests punitively



	A	B	C	D	E
1	**Requestor**	**Requested of**	**By When**	**Status**	**Description**
2	John	Mary	2-May	Complete	Can you provide the first draft of the Vision for the team to review?
3	Mary	Sandra	12-May	Accepted	Will you post the initial use-case model on our project Web site?
4	Tim	Sandra	11-May	Declined	Can you provide me with estimated cost for the Elaboration phase of the project?

Team Activation

- Generate trust
  - Don’t force trust to be earned from the beginning
  - “I trust you”
- Conduct effective meetings
  - Goal-oriented
  - Creative and, yes, even fun
  - Meet when necessary only

Other People Guidelines

- Pay attention to personalities
  - Put people together who can work together
  - Avoid the “partner from hell” pairings
- Recognize and reward
  - Sincerely and often
  - Reward should fit the achievement
    - Be creative
  - Provide for peer recognition

Other People Guidelines

- Encourage “creative, controlled, cowboys”
  - Everyone needs to let their creative selves run sometime
  - Too much creativity can kill the team
- Beware of the touchy-feely monsters
  - Technical people are often not receptive to group hugs

Process

- Process can break people!
- Process must support the people and help them do their job more effectively
- Process must focus on the delivery of the product
- Process should not be static

Process Content And Structure

- Process should address issues at the right level
  - May require different processes (program, product, project)
- Process should focus on delivering software
  - Software is not just code
  - Training, documentation, support

Process Content And Structure

- Process should be as minimal as possible, but no less
  - There is no one-size-fits-all
  - Do not confuse “formal” with “minimal”
    - There is often a relationship, however
  - Minimal artifacts, activities, and overhead

If I don’t do this activity, or produce this artifact, will anything bad happen?

If not, don’t do it!

Process Content And Structure

- The process should provide guidance for all project members
  - Responsibilities
  - Interactions
  - Technical details
- Make sure the process addresses risks
  - If you're not addressing a risk, what are you doing?

Process Enactment

- Make the process yours
  - Configure for your context
  - Apply reuse liberally
  - Approach from an artifact-centric viewpoint
  - Ease-of-use counts
- Start with a proven foundation
  - Best practices
  - Standard framework that can be adapted in a consistent manner

Process Enactment

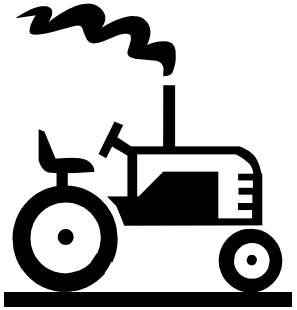
- Adopt process incrementally
  - Assess: identify the pain points and capacity for change
  - Define: customize for your organization or project
  - Deploy: “launch” your process
- Involve the entire team
  - Ownership leads to commitment
  - The team has some of the best ideas

Process Enactment

- Welcome change in the process
  - Review and revise regularly
  - Throw out what doesn't work, add what is necessary
- Support the process with tools
  - Effective automation eases the transition
  - Look for tools for every role
- Don't be too dogmatic
  - Know when to say when

Tools

- Support the best practices in your process
- Support people to employ the best practices more effectively
- The more your tools work together, the more effective you are



Team vs. Individual Tools

- What tools do you care about?
  - Tools that support the process
  - Tools that support the team
- What about individual tools?
  - Text editors
  - Scripts and other aids
- Allow as much individual “comfort” as possible

Tool Selection

- Make sure the tools are worth the effort
  - Identify cost vs. benefits early
  - Measure
- Make sure the tools fit your project
  - Just because you have a tool in your toolbox doesn't mean you have to use it
- Use only the features you need
  - Just because you use a tool doesn't mean you have to use all the features

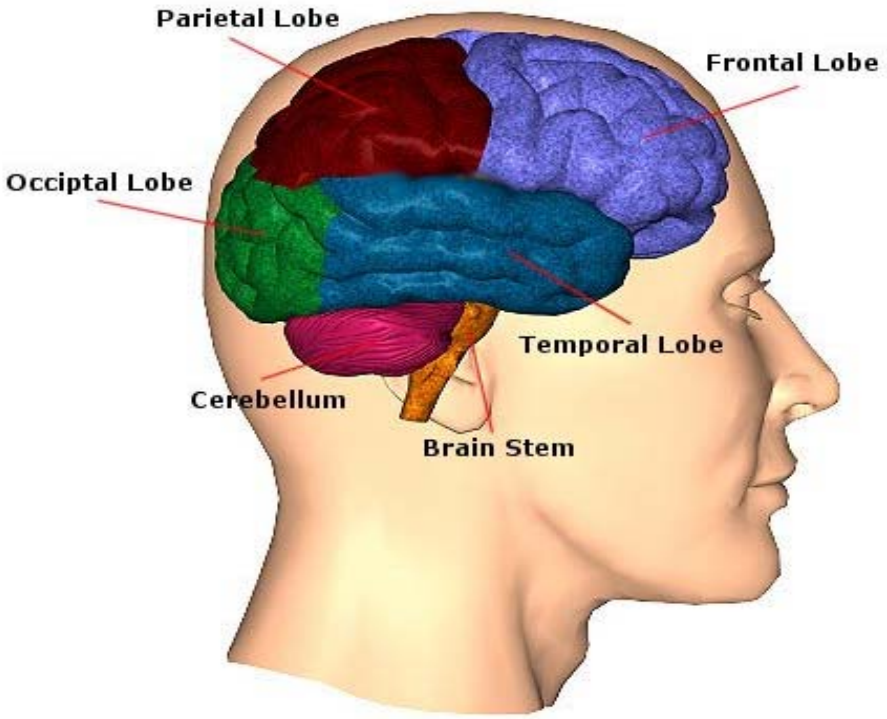
Tool Selection

- Standardize tools as high up in the organization as possible
  - Best for the overall environment
  - Allow exceptions when warranted
- Allow time for tool training
  - Accommodate different learning styles
  - Don’t make this type of learning something people do on their own time

Tool Selection

- Value tool integration
  - Existing integrations are better than ones you need to do
  - Building your own integration is better than none at all

What's Missing?



Brains are required!

Apply common sense liberally

Consider Your Project's Context

- Size
  - Small project – less than 10 people, few month duration
  - Large project – 50 people, multi-year
- Formality
  - Usually small is less formal
  - Consider ultimate goal and exterior constraints

Consider Your Project's Context

- Organizational
  - Team members' familiarity with each other
  - Well-understood alignment with company goals
- Physical
  - Team distribution
  - Physical environment – offices, machines, and so on

The Healthy Project

- Focuses on delivery
- Provides learning opportunities for everyone
- Has just enough process, but no more
- Is supported by a healthy team
  - Respect for each other
  - Recognizes skills and abilities
  - Compensates for weaknesses
- Continues to evolve
- Is one that people want to work on

What Can Go Wrong?

- Symptom: failure to deliver
  - Over-focus on tools (Inspector Gadget strikes)
  - Poor process
    - Consider iterative, incremental development
    - Ineffective communication and understanding of each team member's responsibilities
    - Failure to manage change
  - Team not focusing on the important things
  - Ego wars
    - Don't rule out sabotage

What Can Go Wrong?

- Symptom: Poor quality
  - Tired team
    - Too much to do, too little time
  - Lack of commitment
    - Demands, not requests
    - No learning
  - Ineffective communication
    - Poorly-defined responsibilities
    - Vague goals and vision
  - Improper tool usage
    - No tools
    - Using the tools improperly
    - Using the wrong tools

What Can Go Wrong?

- Symptom: Personnel turnover
  - Lack of freedom
    - Dogmatic application of process
  - Lack of trust
    - No opportunity for the team to get to know each other
  - Little or no learning opportunity
  - Too many stars
    - Ego wars
  - Poor tool support
    - Wrong tools
    - Complex tools
    - Lack of tools
  - Lack of sincere recognition of achievement

Other Things To Look For

- Inmates running the asylum
  - Little or no process
  - Ineffective communication
- Things “missing” (not getting done)
  - Poorly-defined process, lack of goals
  - Too many superstars
- Continual conflict
  - Lack of trust
  - Poorly-defined responsibilities
  - No allowance for disagreements

Other Things To Look For

- Burnout
  - Demands, not requests
  - Too much process overhead
  - Measuring the wrong things
  - Lack of tools
  - Too much to learn
- Too much rework
  - Process is not based upon risk mitigation
  - Lack of prioritization
  - Poor interaction among all team members

Next Steps

- Make your own list of guidelines
  - Use this list as a start
  - Keep what works, get rid of what doesn't
- Map the guidelines to your situation
  - When are the guidelines appropriate?
- Continually review and revise
  - Identify success and failure
  - Communicate your results

Some Resources

- Rational Software Web Pages, www.rational.com
- The Rational Edge, www.therationaledge.com
- The Phoenix Agenda, John Whiteside, 1993
- Adaptive Software Development, James A. Highsmith III, 2000
- Becoming a Technical Leader, Gerald Weinberg, 1986
- Organizational and process patterns, http://i44pc48.info.uni-karlsruhe.de/cgi-bin/OrgPatterns

Thank You

Questions