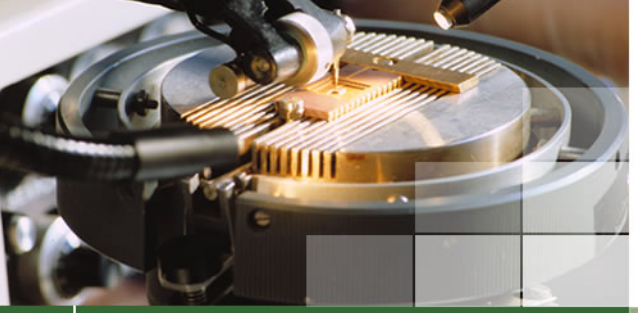


POLITECNICO DI MILANO

Dipartimento di
Elettronica e Informazione

Planning and Managing Software Projects 2011-12

Session 18

Final Stages

Emanuele Della Valle

http://emanueledellavalle.org

Credits

- This slides are largely based on Prof. John Musser class notes on “Principles of Software Project Management”
- Original slides are available at http://www.projectreference.com/
- Reuse and republish permission was granted

Today

- Session 17 review
- Final Steps
- Maintenance
- Post Project Reviews (Post-mortems)
- Success tips
- Capability Maturity Model: CMM
- Final exam review

Project Control

- Planning
- Measuring
- Evaluating
- Acting

Earned Value Analysis

- BCWS
- BCWP
  - Earned value
- ACWP
- Variances
  - CV, SV
- Ratios
  - SPI, CPI, CR
- Benefits
  - Consistency, forecasting, early warning

Final Steps

- Migration
- Roll-Out
- Training
- Documentation
- Shipping Details
- Installation

Migration

- Moving users from existing system to your new one

Migration Plan 1/2

- Includes
  - Description of environment (computers, DBs, interfaces)
  - Description of existing data needed
  - Description of operational constraints
    - when can we move to the new system?
      - Weekends only?
      - Last week of month only?
  - List of affected organizations and contacts
  - Plan of steps to be taken

Migration Plan 2/2

- Does it require a service interruption?
  - If so, when does this happen? A weekend?
- Training?
- Is there a helpdesk?
  - If so, do they have “scripts” or new material?

Migration Strategies – Intro

- Communication with customers is crucial
  - What is happening, when, and why
  - “Why” should remind them of the benefits
  - Not too much detail or too little
  - Where do customers go for more information?
- Minimize intrusiveness
- Find-out about customer’s key dates
  - When does the system absolutely need to be stable?
  - Know about their important deadline dates
  - They must buy-into the approach!

Migration Strategies

1. Flash-Cut
   - Straight-move from old system to new

     A) Immediate Replacement
     - Fastest approach
     - Still want a back-out plan
     - Requires strong planning and testing

     B) Parallel Operation
     - Mitigates risk
     - Parallel to either existing manual or system process
     - Cut occurs once new system “burned-in”

2. Staged
   - Replace one part of existing system at a time

Migration Strategies - Considerations

- Level of business disruption
- Degree of freedom in “production” date
- How much internal opposition to system is there?
  - If higher, perhaps a longer ‘adjustment’ period
- Your comfort level of system quality
  - If questionable, may want to mitigate risk

Cutover: from development to operations

- Criteria: What conditions must be met prior?
- Responsibility: Who decides?
- Operations: Who ‘owns’ it once it’s live?
- Rehearsals: Sometimes used.

Flash-Cut

- Immediate Replacement
  - Ex: new corporate-wide calendaring system
- Requires very careful planning & testing
- Still try to get some users to “try” it first if possible
- Develop a back-out plan

Back-Out Plan

- Especially important for “conversions”
  - Customers already have expectations and needs as defined by their existing system
  - Must be able to restore customer’s service ASAP
- May mean running both simultaneously “just in case”
- Leave it in place for awhile (more than a day!)
- When to back out?
  - Management: sooner
  - Technicians: one-more-fix
  - Set a time limit (ex: 3 hours of start)

Parallel Operation

- Multiple variations of this method
- An “adoption” period
  - See telephone industry with new area codes
  - Both work for a period of time
- Avoid flash-cuts if possible
  - Start with test subjects

Data Conversion

- Quote:
  - If you add a cup of champagne to a barrel of sewage, you'll have a barrel of sewage
  - If you add a cup of sewage to a barrel of champagne, you'll have a barrel of sewage
- Most systems need this step
- Most PMs forget this
- Impacts both completely new and replacement systems
- The "data" often more valuable than the "system"

Data Conversion Areas

- Data Sources:
  - Where does it come from?
  - Do you need to modify data on the way in?
  - Is it accurate?
- Process Controls:
  - Does it happen all at once?
  - How do you guarantee it's been done correctly?
- Completion:
  - How do you handle any 'exceptions'?
  - Do you make backups? Can you restart?

Roll-out

- Create a “Release Checklist”
  - Avoid activities falling through the cracks
  - Example
    - http://www.construx.com/Page.aspx?hid=1216
    - Cached http://emanueledellavalle.org/slides/P&MSP2012_18b_Sample-Release-Checklist-Construx.pdf
  - Activities by Group:
    - Engineering, QA, Documentation, Operations
  - Possibly sign-off signatures
- Roll-out: Must have a plan for the process
  - Often on a given day (ex: a Sat.)

Training

- Often more than just end-users
  - Users
  - Sales & Marketing staff
  - System operators
  - Maintenance engineers (possibly)
  - Sales engineers (possibly)

Documentation

- Must be ready by ship-date
- Final user documentation
- Updates to other
  - Operations documentation
  - Development documentation
  - Sales and marketing material
  - Wed site
  - Test reports

Shipping Details

- Packaging (if commercial product)
- Marketing collateral
- Security mechanisms (if commercial product)
- Licensing

Installation

- Scripts
- Uninstall (if not Web-based)
- If you need to install your software (as on PCs):
  - Don't underestimate:
    - Time this takes to develop
    - Importance of a "first impression"
- Or, if "custom" software you're reselling
  - Installation at site is often a "mini-project"

Maintenance Phase 1/3

- Project management not always carried over
- The “No respect” phase
- Less “glamorous”
  - Lack of enthusiasm
- Pressure to make fixes quickly
  - For “production” problems
- Software can become “hacked” “patchwork” over time
- Finding a support & test platform can be difficult
  - Often the forgotten child until fixes are needed

Maintenance Phase 2/3

- Compare to hardware maintenance
  - Not to keep state same; but changes to state
  - Fixes and enhancements
- Configuration control is very important
  - Fixing the “right” version; tracking branches
- Smaller team
  - Often not a ‘dedicated team’
  - Drawn from developer with other main tasks

Maintenance Phase 3/3

- Contracts, remember those?
  - Always consider the maintenance phase here
  - Often via a “labor hours” contract
    - Time & materials in a “direct” scenario
  - Otherwise via “maintenance contract”
    - Percentage of software license fee
    - Ex: 20% of original cost per year
- Corp. budget if internal projects
  - Often annual/monthly “maintenance” allocations

Project Recovery

- How to save a “drowning project”
- 3 Approaches
  1. Cut the size of the software
  2. Increase process productivity
  3. Slip the schedule, proceed with damage control
- Opportunity for decisive leadership action
- Not a time to ‘just cut corners’
  - Be realistic (not foolish)
- Timing: politically important
  - Not too early, not “too” late

Steps

- Assess situation
  - Is there a hard deadline, what's negotiable, etc.
- Don't do what's been done already
- Ask team what needs to be done

People Steps

- Restore morale
  - Sacrifice a sacred cow
    - Dress code, off-site, catered meals, etc
  - Cleanup personnel problems
- Focus people's time
  - Remove non-essential work

Process Steps

- Fix classic mistakes
  - Inadequate design, shortchanged activities, etc?
- Create “Miniature Milestones”
  - Small (in day(s)), binary, exhaustive
  - Boosts morale: getting things done!
- Track progress meticulously
- Recalibrate after a short time
- Manage risk painstakingly

Product Steps

- Stabilize the requirements
- Raise the bar on change requests
- Trim the feature set
  - Determine priorities, cut the low ones
- “Take out the garbage”
  - Find error-prone modules; re-design
- Get to a known, stable state & build from there

Post Project Reviews (PPR)

- a.k.a.
  - Lessons Learned Review
  - Postmortem
  - Post Project Analysis (PPA)
  - Post Performance Analysis
- Focused on: Process not People!
  - Potentially a finger-pointing, blame-game exercise

Steps

- Email team to schedule meeting
- Use a Survey Form to gather initial feedback
  - http://emanueledellavalle.org/slides/P&MSP2012_18c_LBH_Post_Project_Review_Template_TMP.doc
- Ask them to collect all potentially relevant data
  - Dimensional project data work products: size, qty, etc
  - Change requests
  - Time and effort data
- Conduct meeting
  - Collect data and feedback, discuss
- Summarize in a PPR report

Success Metrics

1. On schedule
   - Requires good: plan; estimation; control
2. Within budget
   - Again: planning, estimation & control
3. According to requirements
   - Importance of good requirements
   - Perception & negotiation critical

Success Rates

- By Industry
  - Best: Retail
    - Tight cost controls in general
  - Worst: Government
    - Least cost controls
- By Size
  - Smaller is better: cost, duration, team
- Stats
  - http://www.ambysoft.com/surveys/success2007.html
  - http://www.ambysoft.com/surveys/success2008.html
  - http://www.ambysoft.com/surveys/success2010.html
  - http://www.ambysoft.com/surveys/success2011.html

You are not Santa Claus

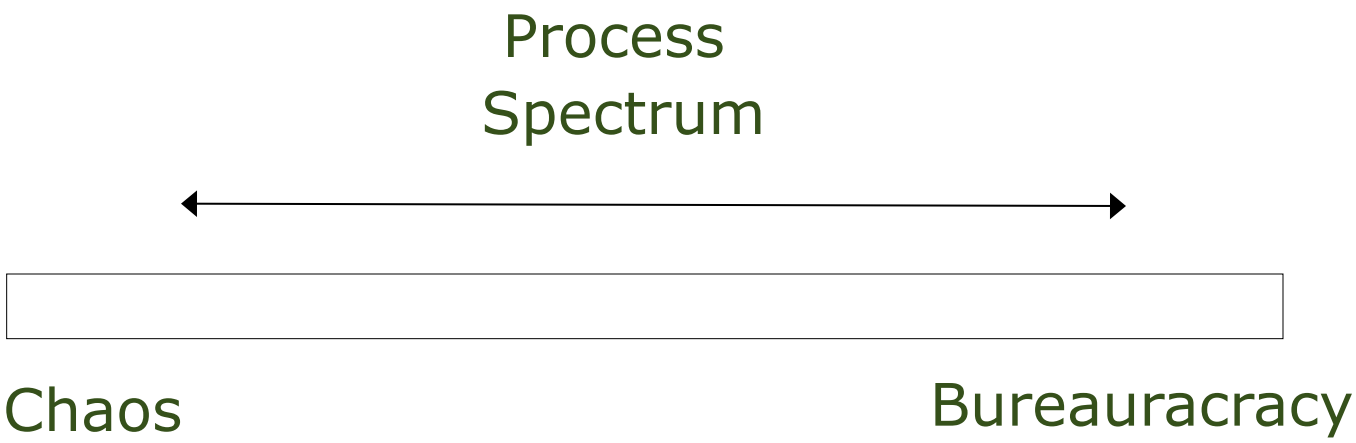
- Learn to say “No”
  - Be polite but firm
- The Value of Versions
  - “We will put that in phase 2”
- An Ounce of Prevention

Think Small

- Keep requirements tight & focused
- One milestone at a time
- Smaller, incremental chunks
- As simple as possible but no simpler

Process Spectrum

- Too much medicine can kill the patient



- Balance is crucial

Paralysis

- Analysis Paralysis
  - Over-process
  - Nothing gets finished
  - 65% of software professionals have experienced this
- Paralysis Paranoia
  - Fear of over-process = process avoidance

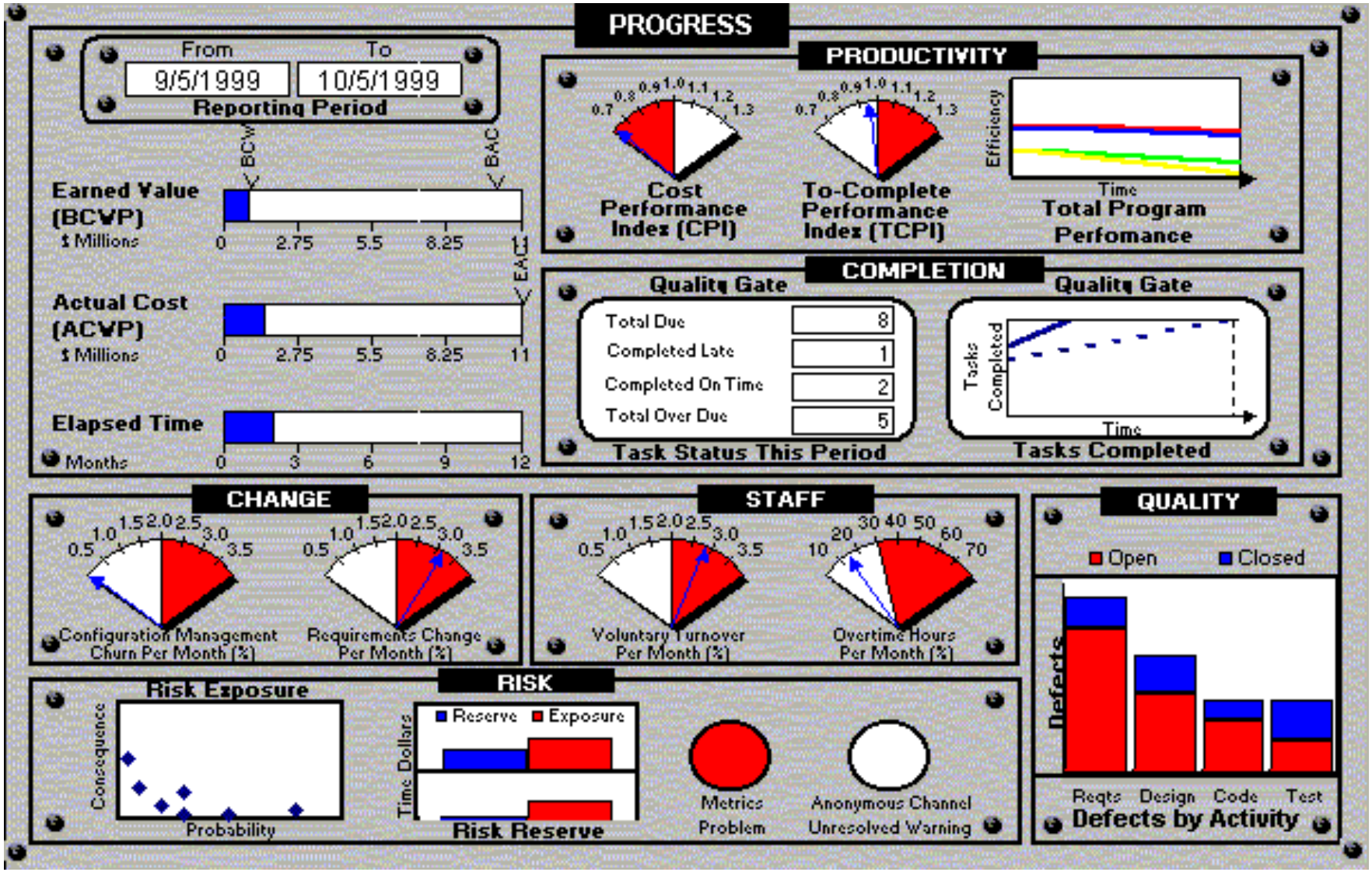
Management by Walk About

- Shows your actually involved day-to-day
- Recognizes individuals may say more 1-on-1
- Allows spontaneity
- Finds personnel problems sooner

Delegate

- Don't be a "Control Freak"
- You need to be the "hub" but not everything

Tools - Project Control Panel



Capability Maturity Model: CMM

- A software process framework
- “Process determines capability”
- 5 ‘maturity’ levels
  - ‘Evolutionary plateaus’ to a mature software process
- Each level has its own goals
- Organizations can be ‘certified’
  - Later to be used as a marketing or validation tool
  - http://www.itil-officialsite.com/home/home.asp
- Links:
  - SEI - http://www.sei.cmu.edu/
  - Diagram - http://www.sei.cmu.edu/cmm/

Levels 1/2

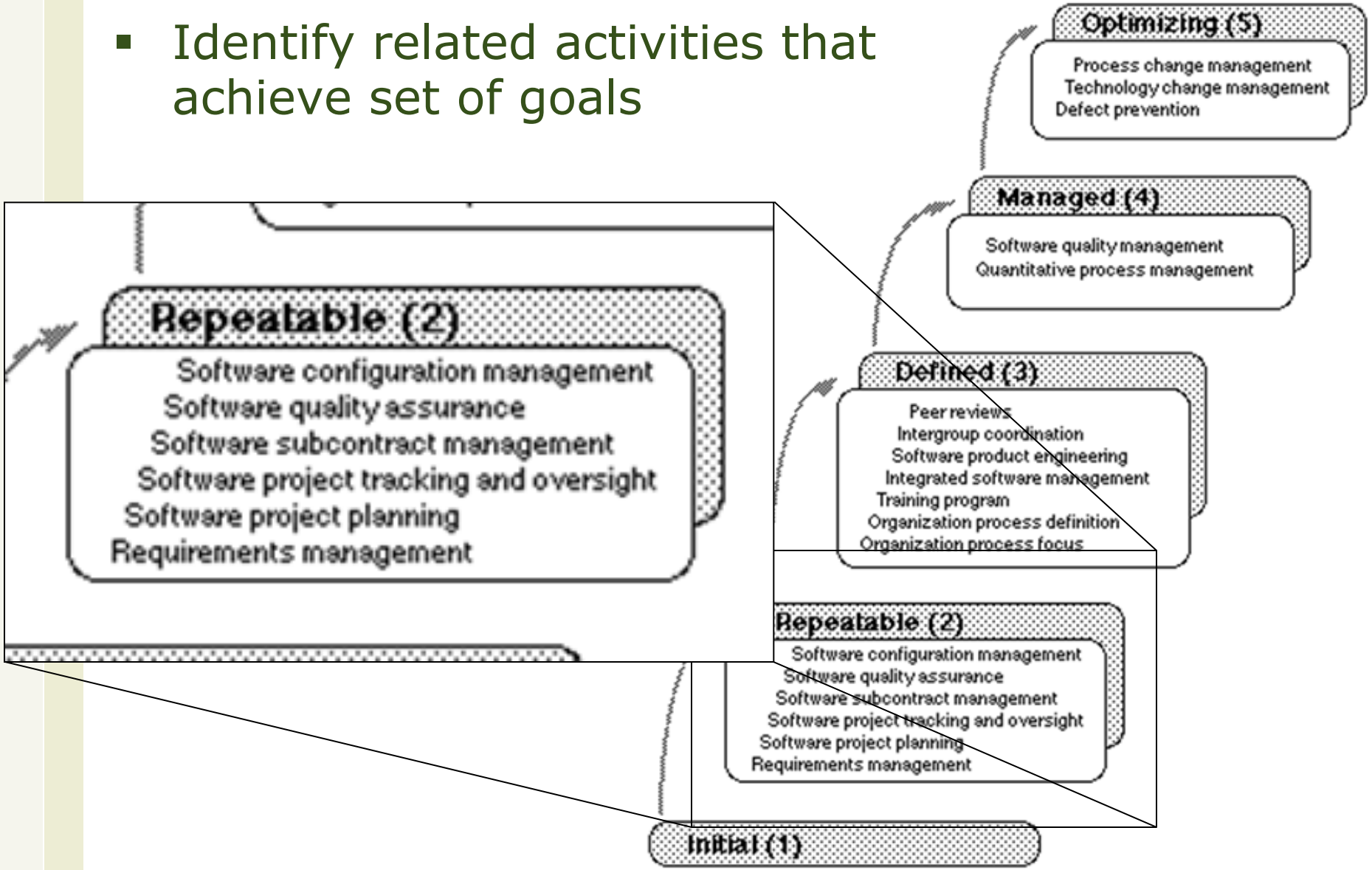
1. Initial
   - 'Ad hoc' process, chaotic even
   - Few defined processes
   - Heroics often required here
2. Repeatable
   - Basic PM processes
     - For cost, schedule, functionality
   - Earlier successes can be repeated
3. Defined
   - Software & Mgmt. process documented
   - All projects use a version of org. standard

Levels 2/2

4. Managed
   - Detailed metrics of process & quality
   - Quantitative control
5. Optimizing
   - Continuous process improvement
   - Using quantitative feedback

Key Process Areas

- Identify related activities that achieve set of goals



Continuous Software Process Improvement (SPI)

- Herbsleb, 1994, "Benefits of CMM-Based Software Process Improvement"

Measure Category of SPI	Range	Median	Number of Data Points
Years engaged in SPI	1 - 9	3.5	24
Yearly cost of SPI per software engineer	$490 - $2004	$1375	5
Productivity gain per year	9% - 67%	35%	4
Early defect detection gain per year	6% - 25%	22%	3
Yearly reduction in time to market	15% - 23%	19%	2
Yearly reduction in post-release defect reports	10% - 94%	39%	5
Business value (savings/cost of SPI)	4.0 - 8.8	5.0	5

Who needs a CMM certification?

- India has more CMM level 4 & 5 companies than any other country
  - Why is that?

Questions?