Dividing and Conquering Those Lingering Bugs

- or -

Organize Your Socks!

Barry Mirrer
Director of Quality
OUTCOME
Cambridge, MA

Finding a Pair of Socks

- Look at your socks. Are they in a drawer? What's in there? Do you have socks that you'll never wear?
- All the "extra" socks that you need to go through every day get in the way of finding a GOOD pair of socks, slowing you down and making you less EFFICIENT.

Step 1 - Assessment

- Where are the socks?
- What do you have?
  - Nice, neat pairs of socks
  - Singles
  - Socks with holes
  - Sock you wouldn't be caught dead in
  - Dirty socks
  - Old ticket stubs, keys, knick knacks

Step 2 - Planning

- Decide which socks to keep
- Decide what to do with the remaining socks...
  - Put aside singles for 6 months; if you don't find the match, throw it away.
  - Sew or darn socks with holes
  - Throw out socks from high school
  - Wash the dirty socks
  - Give away psychedelic socks

Step 3 - Infrastructure

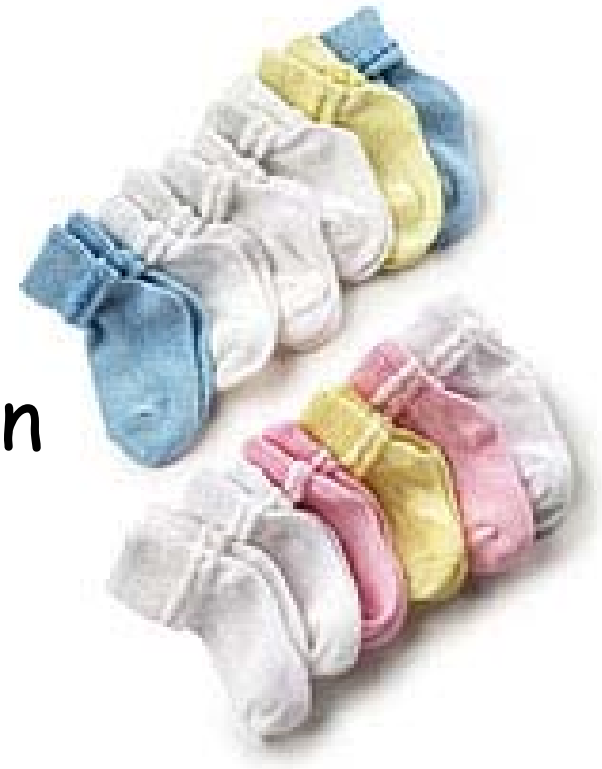
- Get a sock drawer
  - Buy one (or more!)
  - Build them yourself
- You may want several drawers for different type of socks, depending on your plan.
- Buy a needle and thread
- Gas up the car

Step 4 - Execution

- Put aside the "keepers"
- Find a place for the singles
- Learn to sew and schedule time to repair socks
  - … or throw them away and buy new ones
- Put dirty socks in the wash
  - Then fold them and put them away!
- Schedule trip to the thrift shop

Step 5 - Celebration

- You did it!
- Revel in the feeling
- Feel the endorphins
- Commit to perpetuation
- Celebrate!



Now, what about those bugs?

- Do you have a database?
- Problems with a disorganized or overstuffed database
  - Inefficiency
  - Morale
  - Tracking and Reporting
  - Space (disk space, backups, reports)
  - Finding the diamonds

Step 1 - Assessment

- What's the vision?
- Where are the bugs?
- What do you have?
  - Old ones
  - New ones
  - Enhancement requests
  - New functionality
  - Process issues
  - Suggestions
- What's the goal?
  - Duplicates
  - Reminders
  - Fixed bugs
  - Bugs on unsupported products
  - Unprioritized bugs

Step 2 - Planning

- Break the problem into chunks.
- This is usually based on:
  - Age
  - Status
  - Type of problem
  - Release
  - Product
  - Priority
  - Areas of expertise

Step 2 - Planning (cont.)

- Decide what needs to be kept. Make sure these are not eliminated.
  - Example: bugs reported in last 6 months
  - Be honest
  - Be realistic based on the resources you want to expend on this effort
    - Rule of thumb: On average, one engineer can fix one software bug in one week.
    - Estimate your costs from this number

Step 2 - Planning (cont.)

- Chunk the remaining bugs
- Are there chunks that will be easy to eliminate?
  - Examples:
    - Products that are no longer supported.
    - Low priority problems over 2 years old.
    - Duplicates.
    - One-time crashes that are over a year old.
    - Internally-found problems over 18 months.

Step 2 - Planning (cont.)

- Fit every problem into a category
- Associate an action with each category
- Do this with the buy-in of affected groups such as
  - Customer Support
  - Software Test
  - Documentation

Step 2 - Planning (cont.)

- Examples of actions:
  - Close
  - Review with Support before closing
  - Reproduce in the current release*
  - Assign to maintenance group
  - Commit to a maintenance release
  - Assign to engineers (only if the assignment includes a committed timeframe and resources)

Step 3 - Infrastructure

- Database structure
- Categories
  - Update if necessary to reflect Plan
- Access to database and functions
- Assignment of resources
  - Goals and incentives
  - Is training necessary?
- Relationships and buy-in
  - Sharing the plan

Step 4 - Execution

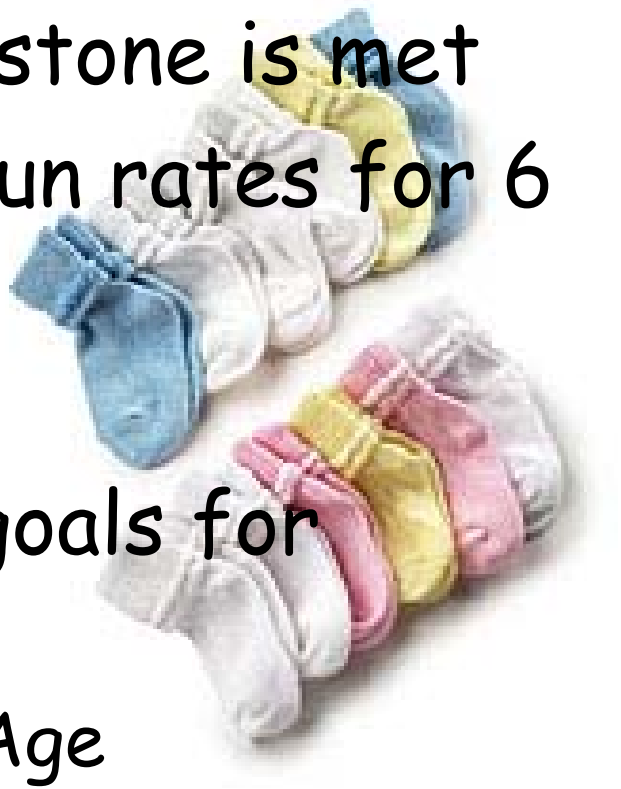
- Communicate vision and goals
  - Include interim goals
- Assign resources
  - Make sure people have proper skill sets
  - Train as necessary
  - Put incentive and reward system in place
- Gain cooperation of affected groups
- "Mass update" DB if appropriate

Step 4 - Execution (cont.)

- Monitor progress against plan
- Communicate progress
  - Charts
  - Reports
- Checkpoint plan at regular intervals
  - At least once every quarter

Step 5 - Celebration

- Celebrate as each milestone is met
- Celebrate successful run rates for 6 months or a year.
- The job is never done
- Set and communicate goals for continued control
  - Bug Count vs. Average Age



Real-Life Examples - 1

- >2000 Problems Reduced to 600
  - Obsolete products eliminated
  - Problems older than 12 months closed with Customer Support approval
  - Development "blitz"es
  - Reproduce problems in current version of software, or close them if fixed
  - Hired dedicated staff
    - Senior-Level Engineers

Real-Life Examples - 2

- 1200 Problems Reduced to 400
  - Obsolete products eliminated
  - Older low priority problems closed as "Will Not Fix" w/Cust. Support approval
  - Engineers focused on fixing PRs ("blitz")
  - Non-defect problem reports such as Enhancements were separated out
  - Hired dedicated staff
    - Third Party w/product expertise

Help and Ideas

- Eliminate PRs on unsupported products
  - And/Or categorize them as "known problems that will not be fixed"
- Eliminate PRs on unsupported releases
  - May reproduce on the current release and update the PR or close as fixed

Evaluate Your Data

PRIORITY	ACTION
Urgent	Always leave Open
High	Re-evaluate bugs over 6 months old
Normal	Close bugs >12 mos old after CS review
Low	Close bugs >24 mos old after CS review
Enhancements	Review w/Marketing

Categorize Non-Defects

- Enhancement Requests
  - Evaluate during product planning
- Open for Information/Tracking purposes
  - Evaluate at regular intervals (6 months)
- Internal Infrastructure Issues
  - Evaluate at regular intervals (6 months)

Categorize Defects

Internal and External are Treated the Same

- Urgent
  - Close in 1 month; update every week
- High
  - Close in 8 months; update every release
- Normal
  - Close in 12 months; update every release
- Low
  - Close in 16 months; update every release

Communication to Customers

- Customers can take good new and they can take bad news. But they can't take NO NEWS!
- Use this process as an asset
- Get back to customers sooner
- Use customers as a partner
- Reverse your decisions if necessary
  - Use customer feedback

Status Quo Systems

- Ideally, you should be able to evaluate and assess a problem when it arrives
- Determine a static run rate using input and output bug rates.
  - Barry's un-validated rule of bugs...
- Count on increased efficiency
- Monitor the system, report monthly, quarterly, annually.

Helpful Hints

- "Someday" bugs
- A word on milestones
  - The garage and heavy lifting
- Communicate, communicate
  - The 2 rules of communication

Other Information

- Web site for lost socks:
  http://www.hoosiertimes.com/cgi-bin/socks