Effective Bug Management
Challenges and Best Practices

Michael Stahl

SW Validation Architect, Intel

michael.stahl@intel.com

(c) Michael Stahl, 2009. All Rights Reserved

Our jokes may be different…

…but our bug management problems are probably like everyone else's

(c) Michael Stahl, 2009. All Rights Reserved

Disclaimers

- Names and brands referenced herein may be claimed as the property of third parties
- The views expressed in this presentation are solely my own, and do not in any manner represent the views of my employer
- Information in this presentation is provided “AS IS” without any warranties or representations of any kind

(c) Michael Stahl, 2009. All Rights Reserved

Agenda

- **Heaven:**
  - Bug Management in a Perfect World
- **Earth:**
  - Bug Management in Real Life
- **Heaven on Earth:**
  - Improving the Bug Management Process
- **Tips**
  - Stuff that worked for us
- **Summary**

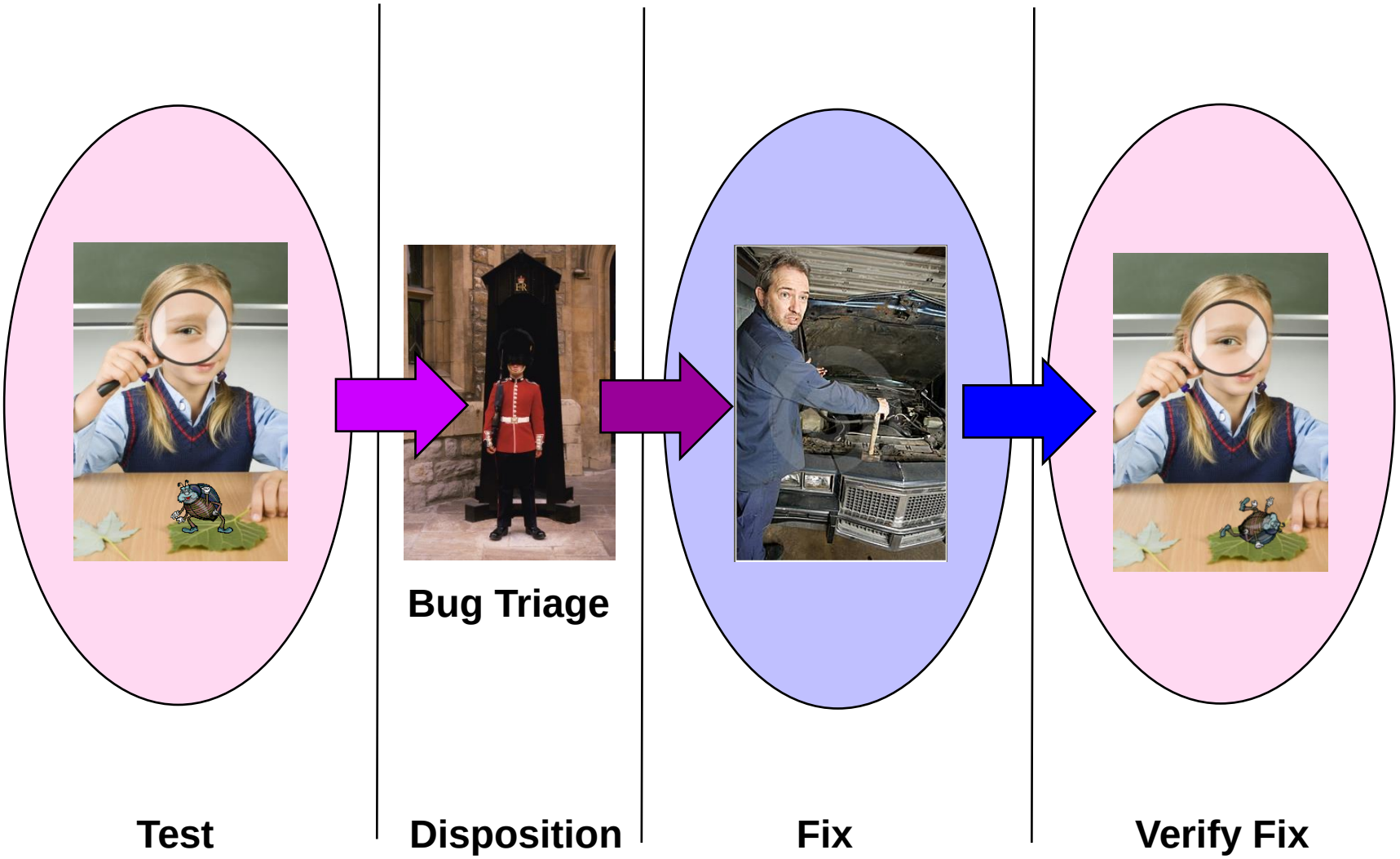
(c) Michael Stahl, 2009. All Rights Reserved

Heaven

Bug Management in a Perfect World

(c) Michael Stahl, 2009. All Rights Reserved

Bug Management in a Perfect World



(c) Michael Stahl, 2009. All Rights Reserved

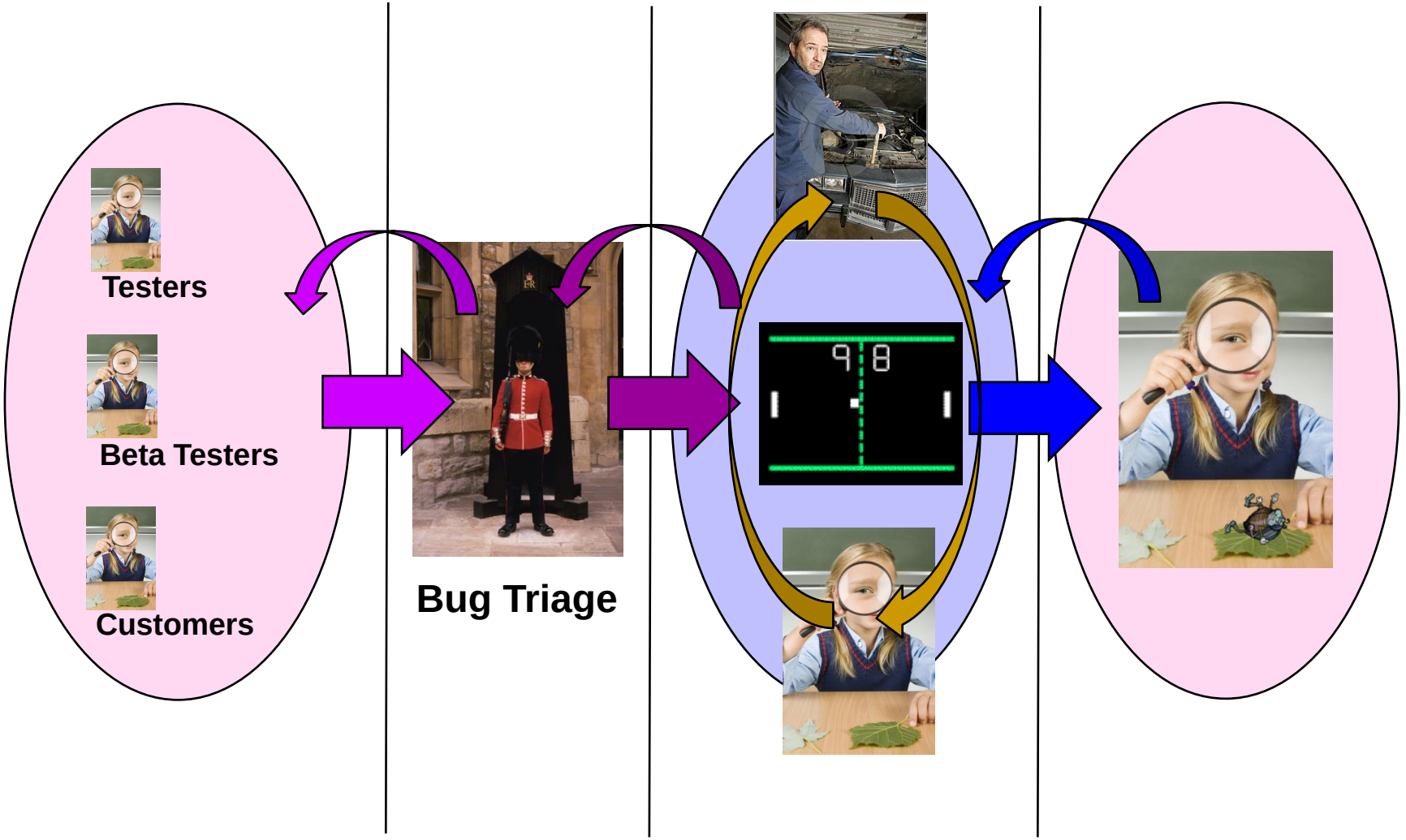
Earth

Technical, People and Organization aspects:

Bug Management in Real Life

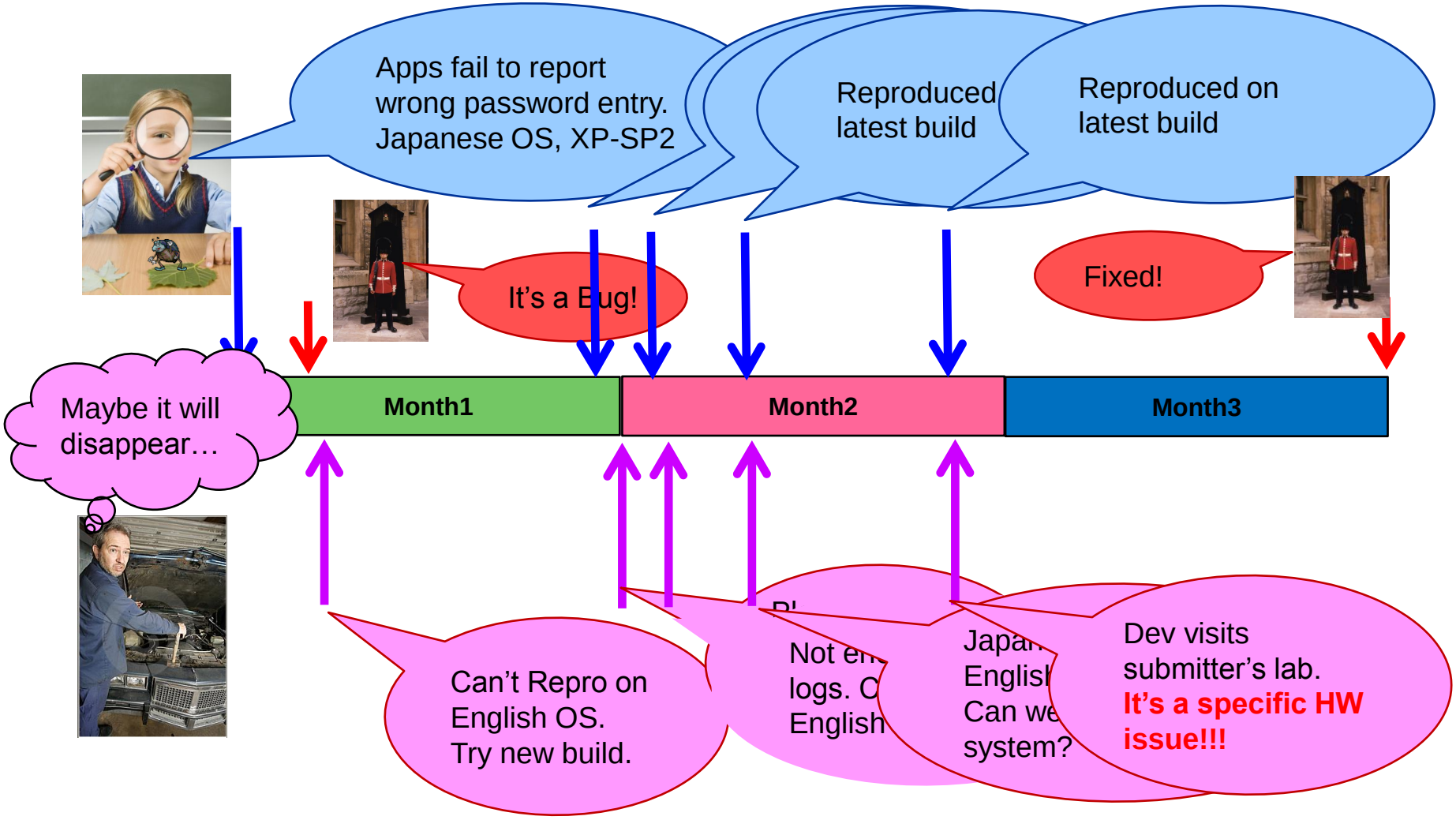
(c) Michael Stahl, 2009. All Rights Reserved

Technical Issues:
Bug Management in a Less-Than-Perfect World



(c) Michael Stahl, 2009. All Rights Reserved
Animated GIF (by permission) www.oblivion-graphics.com/home/video.htm

The Ping Pong Game – A real-world example



(c) Michael Stahl, 2009. All Rights Reserved
Animated GIF (by permission) www.oblivion-graphics.com/home/video.htm

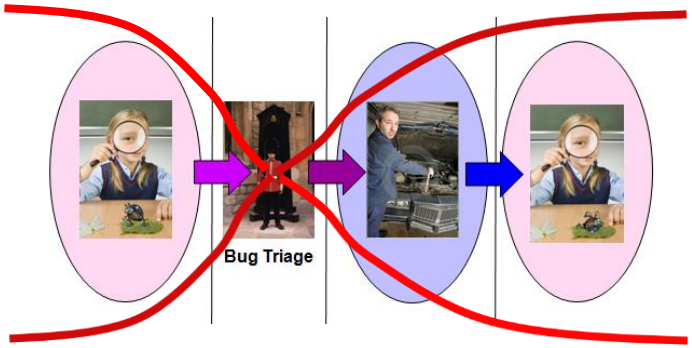
Bug Triage is a Bottleneck

The Technical side

- **Severity guidelines**
  - Ambiguous by definition
  - Subjective, yet used as hard rules
  - What's the Priority?
  - Per product? Per milestone?
  - Team diversity fosters disagreements

The Political side

- **High Severity**
  - Black eye to the Dev team; Badge of honor to Test team
  - Higher chance of getting fixed
  - Has program implications
    - A Show stopper is… a Show Stopper

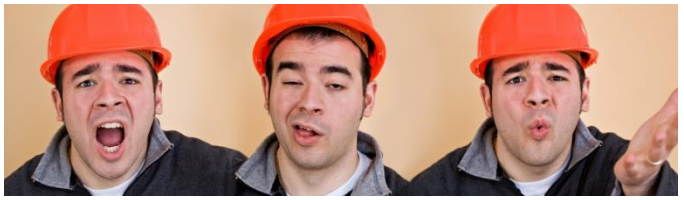


Bug Triage may become an excuse not to deal with bugs

"It's not reviewed yet"

(c) Michael Stahl, 2009. All Rights Reserved

People issues: Bug Triage Meeting



- **Unprepared attendees**
  - "Let me get back to you on this one…"
- **Getting side-tracked**
  - Arguing the bug – not it's Impact
  - Solving the bug in real time

- **Inefficient decision-making process**
  - In many cases: by consensus
- **People get unreasonable**
  - The strong voice (literally!) prevails
  - The Fanatic Tester – everything is a Show Stopper
  - The Push-Back Developer – It's not a Bug!

- **Exacerbated by**
  - Logistics
  - Cultures

(c) Michael Stahl, 2009. All Rights Reserved

Organizational issues

- **Who owns Bug Management?**
  - Validation? Development? Program Office? Marketing?
- **Resource Constraints & Attitudes**
  - **Developers**
    - Completing features is higher priority
      - Higher visibility on the project dashboard
  - **Testers**
    - Role ends with finding the bug
    - Finding more bugs is higher priority over getting bugs fixed

(c) Michael Stahl, 2009. All Rights Reserved

Organizational issues

- **Who owns Bug Management?**
  - Validation? Development? Program Office? Marketing?
- **Resource Constraints & Attitudes**
  - **Developers**
    - Completing features is higher priority
      - Higher visibility on the project dashboard
  - **Testers**
    - Role ends with finding the bug
    - Finding more bugs is higher priority over getting bugs fixed
    - **Throw over the wall!!!**

What is hateful to you, do not to your fellow man

Talmud, Shabbat 31a

(c) Michael Stahl, 2009. All Rights Reserved

Heaven on Earth

Improving the Bug Management process

(c) Michael Stahl, 2009. All Rights Reserved

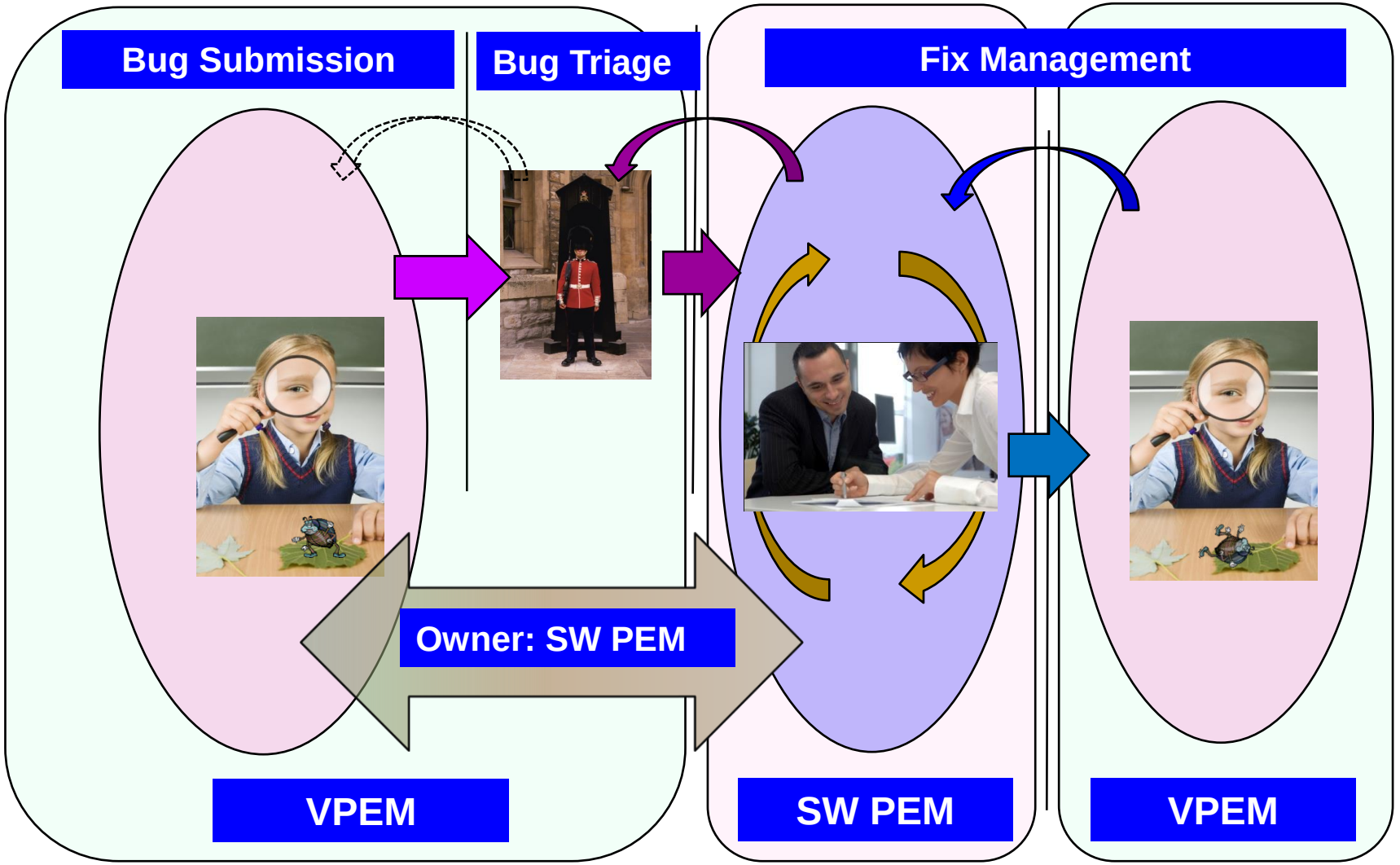
Define the Process; Change the Attitude

- **Make Bug Management a clearly defined Process**
  - Bug Submission
  - Bug Triage
  - Fix Management
- **Change the attitudes**
  - Testers:
    - Getting a bug fixed is **as important** as finding it
    - Sometimes, a bug can't get fixed without the tester's involvement
  - Developers:
    - A submitted bug is (usually) a real one
    - Don't wait for Bug Triage's approval stamp
    - **YOU** are the bug owner
  - PEMs:
    - Internalize that from a certain point, Bug Management is the **critical flow** of the program

"Attitude is a little thing that makes a big difference."

- Winston Churchill

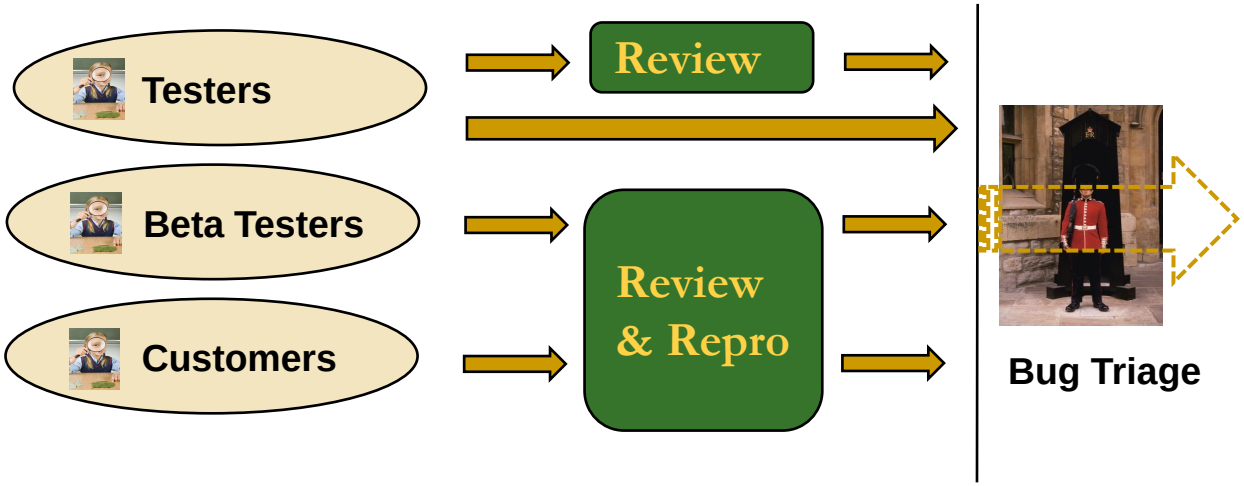
Bug Management Process



Intel Dictionary:
PM: Program Manager **PEM:** Program Engineering Manager; we have: **SW PEM**; **VPEM** (Validation PEM)

Bug Submission Quality

- Define bug templates
- Define mandatory fields and data
  - Keep choose lists updated
- Provide tools for easy collection of the information
- Train, train and re-train
- Review reports of new testers until they can be trusted
- Always review Beta and Customers bugs



Strive to achieve "Workable Bugs"

Bug Triage Process - Notes

- **Owner: Validation. Attendees:**
  - Owns the meeting, the process and the “DB Freshness”
- **Charter: Escalation path for bug disposition**
  - Most bugs are straight forward and can skip the Bug Triage involvement
  - Remove the bottleneck: Dev/Test interact from submission
- **Preparation is key**
  - Bug-list for discussion – sent ahead of the meeting
  - Attendees mark items for discussion
- **Keep tight control of the meeting**
  - Keep a fast pace. Nip irrelevant discussions in the bud
  - Address unreasonable behaviors offline
- **Publish Severity guidelines**
  - Emphasize the word “guidelines”…

(c) Michael Stahl, 2009. All Rights Reserved

Bug Triage Process (Cont.)

- **Adopt an efficient decision making process**
  - Benevolent Dictatorship
  - Escalation path is the Program Manager
- **Avoid setting Priority**
  - Exception: Special cases (Low customer impact; High operational impact => high priority)
- **Avoid being pulled into Bug Management**
  - It's a Program Management activity
- **Deal with the environment**
  - Get a good phone; use screen-sharing software; learn the culture
- **Optional: Monitor Bug Management process metrics**
  - e.g. Time to fix, bug age, time to fix-verify, time in "waiting for inputs"; ping-pongs; false-alarm bug reports

(c) Michael Stahl, 2009. All Rights Reserved

Tips

Stuff that worked for us

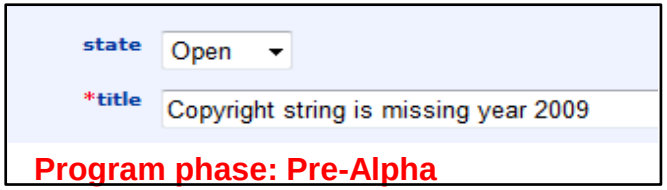
… but will it work in Europe?

(c) Michael Stahl, 2009. All Rights Reserved

Bug Triage tips: Local or Global Severity?

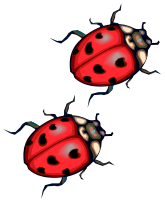
Problem:

- Severity is read as Priority and is a Release Criteria
- What's the severity of this bug?
  - Can't ship without fixing. So: Show Stopper!
  - Can wait till Gold. So… Medium?...



- How do you assign a Show Stopper, but mark it for later fix?
  - Priority? (Show Stopper / Low Priority)
  - Severity? (e.g. Medium for Alpha, Show Stopper for Beta)
  - Milestone Release Criteria must be relaxed to allow delay of non-urgent Show Stoppers

Our Solution:



- Severity, Milestone are set "Globally" – by Bug Triage
  - Priority set by SW PEM and re-evaluated after each milestone
  - Release criteria: Zero SS for the milestone, X SS for the program

(c) Michael Stahl, 2009. All Rights Reserved

Bug Triage tips: Non-reproducible bugs

Problem:

- Can't fix them, can't close them, and can't decide the severity…

  …but we want them reported!

Solution:

- Status = "Suspended"
- Num_reported_instances = 1, and incremented per bug occurrence
- IF num_reported_instances > **X** => Status = "New Bug"
- IF State = "Suspended" for **Y** weeks => Status = "Tabled" (bug is closed)

(c) Michael Stahl, 2009. All Rights Reserved

Fix Management: Daily Ops Meeting

An option for effective Fix Management activities

- **Frequent, Operation-focused meeting**
  - Led by SW PEM, attended by Dev leads, Test lead
  - Sets immediate priorities for the day
- **Frequency and length vary**
  - Typically: Daily or once in two days, for 30-60 min.
- **Works well for short bursts of focused activity**

(c) Michael Stahl, 2009. All Rights Reserved

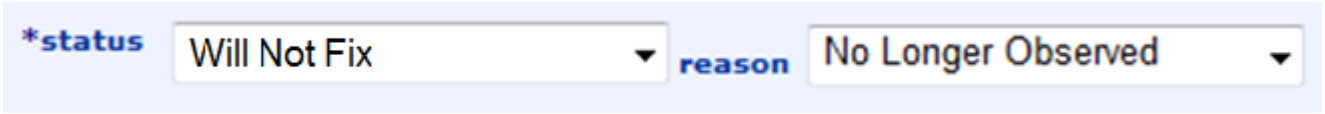
Fix Management tips: Bug "Disappearance"

Problem:

- A bug is reported, with supporting data
- Next build, it is gone
- What has changed?
  - A lot!
  - Dev don't have the bandwidth to investigate what caused the fix
  - Test don't agree to "Can't Reproduce"

Solution:

- Add "Reason" field to the system; value= "No Longer Observed"



(c) Michael Stahl, 2009. All Rights Reserved

Fix Management tips: Late fix scramble

Problem:

- Development is busy with new functionality, and delays bug fixes
- Fix verification (and resulting bugs) are pushed towards the deadline
- Risk of missing the Release Criteria

Solution:

- Monitor “projected opens” VS Release criteria
- Alerts development early how bad their backlog is

Panic early and avoid the rush

- ?

(c) Michael Stahl, 2009. All Rights Reserved

Fix Management tips: Avoid late-fix scramble

(c) Michael Stahl, 2009. All Rights Reserved

Summary

- **Bug Management:**
  - Simple in theory, complicated in practice
  - Three main parts:
    - **Bug Submission process**
    - **Bug Triage**
    - **Fix Management**
  - Impacted by Organizational, People and Technical issues
- **Defined, agreed-upon process is key**
  - Training is a must
  - Ongoing effort to keep up
- **Specific issues call for innovation**
  - New bug states: “Suspend”, “Fixed Indirectly”
  - Forward-looking reports

(c) Michael Stahl, 2009. All Rights Reserved

Acknowledgement

- **Menahem Zukerman (Program Manager, Intel, Israel)**
  - Menahem lead a task force to analyze current practices and suggest new process for Bug Management. Some of the insights in this paper are due to his work.

… and Thank You!

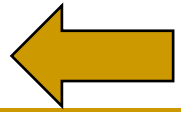
(c) Michael Stahl, 2009. All Rights Reserved

Backup

(c) Michael Stahl, 2009. All Rights Reserved

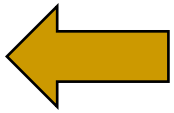
Bug Triage Team Members

- **Bug Triage Lead (VPEM)**
- **Engineering Leads**
  - SW PEM
  - Development reps (per component: e.g. Application, FW, Driver)
- **Validation & Test Leads**
- **Marketing**
  - Customer Application Engineers Rep
- **Optional or “As needed" Attendees:**
  - Tools Engineering leads
  - Localization & Documentation Engineering leads
  - Project Program Manager
  - Other Technical Leads (Mfg, Planning, dependent projects)
  - Other submitter’s of bugs (i.e. as needed once a bug is submitted)
  - QA/QRE



(c) Michael Stahl, 2009. All Rights Reserved

Bug Template



Title:

Summary:

Steps to Reproduce:
1.
2.
3.

Expected Results:

Actual Results:

Recovery steps:

Setup:

- **Name and type of machine:**
- **Processor(s), speed, memory, BIOS version:**
- **Release Version:**
- **Clean System (y/n)?:**

Additional Info:

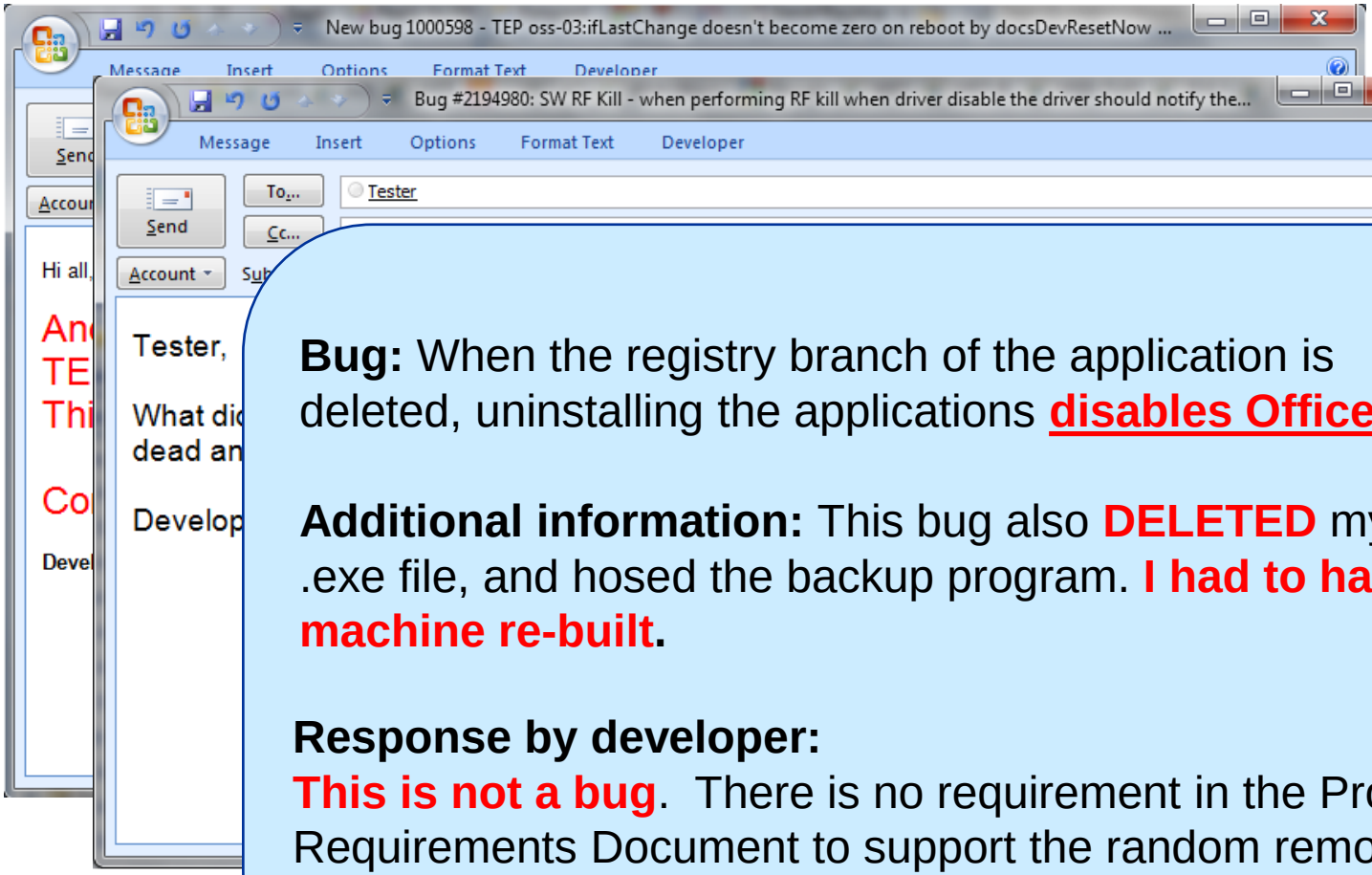
- **Reproducibility (Frequency):**
- **Log Data, Screen captures, Video**
- **Configurations:**
  - **Different OS -**
  - **Different Machine/Platform -**
  - **Different card/sku -**
  - **Different environment variables -**

Suggested Severity:

Reason for Suggested severity (include User Impact!):

(c) Michael Stahl, 2009. All Rights Reserved

Dev/Test examples



Bug: When the registry branch of the application is deleted, uninstalling the applications **disables Office 2003**

Additional information: This bug also **DELETED** my VPN .exe file, and hosed the backup program. **I had to have the machine re-built.**

Response by developer:
This is not a bug. There is no requirement in the Product Requirements Document to support the random removal of registry keys.

(c) Michael Stahl, 2009. All Rights Reserved