Optimizing the Defect Lifecycle – with Resolution

Silverpath Technologies Inc.
Trevor.Atkins@silverpath.com

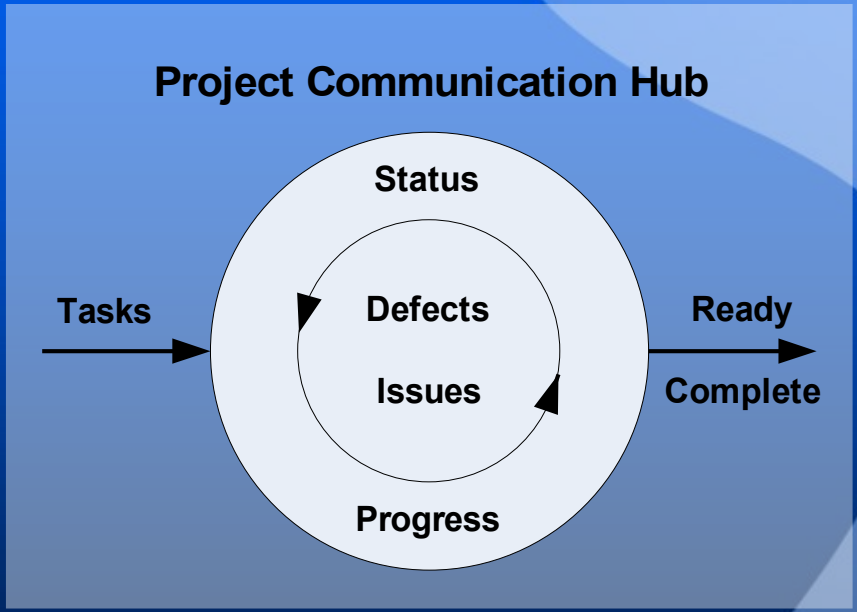
Opportunity for Improvement

- According to a National Institute of Standards and Technology study:
  - Software errors cost the U.S. economy an estimated $59.5 billion annually, or about 0.6% of the GDP
  - 80% of the software development costs of a typical project are spent on identifying & fixing defects
  - About 1/3 of these costs, or an estimated $22.2 billion annually, could be eliminated by an improved testing infrastructure

- NIST 2002-10 Report

The Defect Lifecycle

- A core process in any company that produces software
- Often channels the majority of the project team's interactions
  - Challenge: "Teflon" culture and/or excessive dialogue or "churn" around addressing issues
  - Challenge: Decisions about functionality made by team members who are not in that role



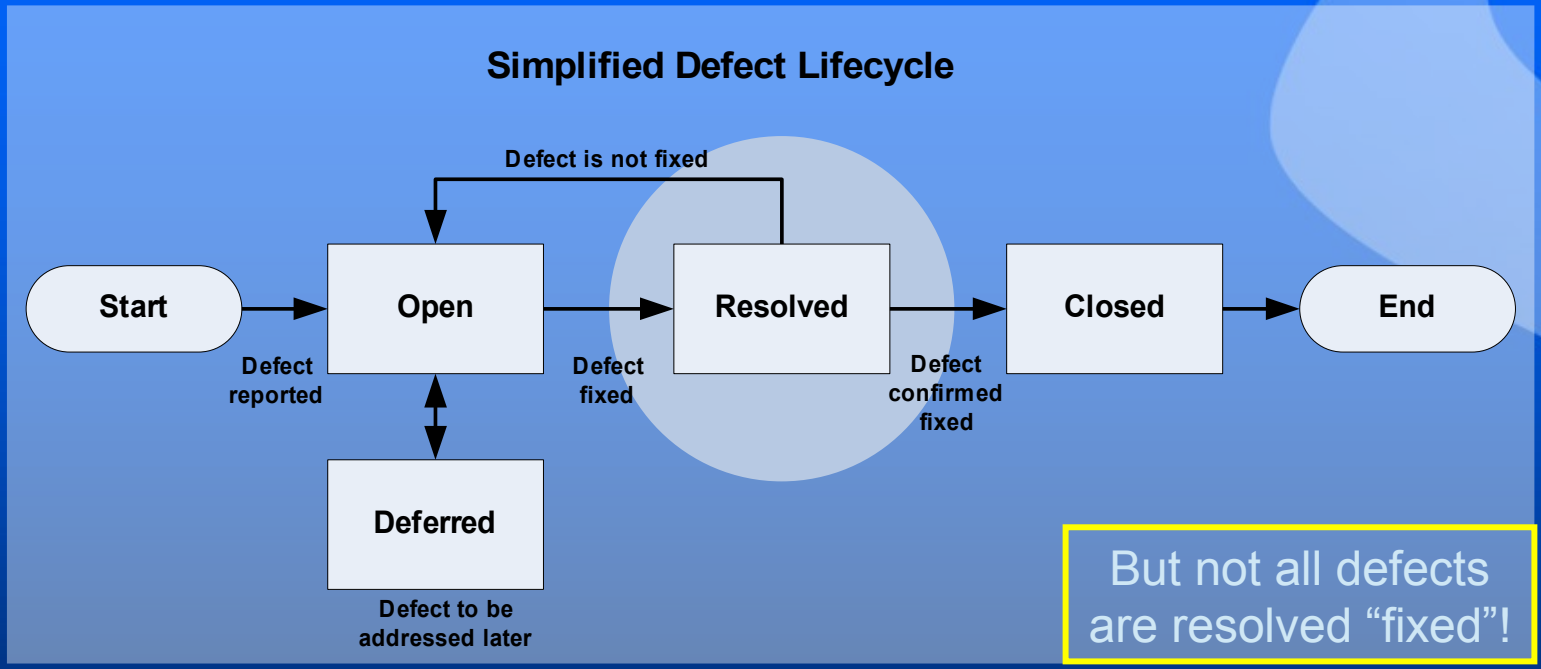
- Makes for an important area where a little effort can give big returns…

So, You Have Found a Bug

- What happens next?
  - How is that quality report handled?
  - How is it investigated?
  - How is it determined to be resolved?

- How do we manage the intense communication that occurs as a project enters into its testing cycles
- How do we track & lower the costs of these activities?

Leverage Defect Resolution



- Obtain an accurate picture of the defect counts by providing the right set of choices for resolution
- Drive ownership & closure of the issue through process automation

Example Defect Resolutions

Silverpath Technologies

- **Fixed:** The programmer says it's fixed. Check it
- **Need Info:** The programmer needs more info about the bug. Elaborate, talk to them
- **Duplicate:** This is a repeat of another bug report. Cross reference it on this report
- **3rd Party:** This bug lies in code outside of the software under test
- **Deferred:** It's a bug, but it will be fixed later. Reopen at the scheduled time. Aka: Postponed
- **Cannot Reproduce:** The programmer can't make the failure happen. Confirm it still happens, add details, notify the programmer. Aka: Not Repro
- **By Design:** The program works as it's supposed to. Get confirmation, update your tests. Aka: As Designed
- **Spec Issue:** The program works as documented, but maybe the requirements are wrong, incomplete, ambiguous, etc. Appropriate new issue opened after discussion & decision is made. Aka: Requirements Issue, Enhancement, Feature Request

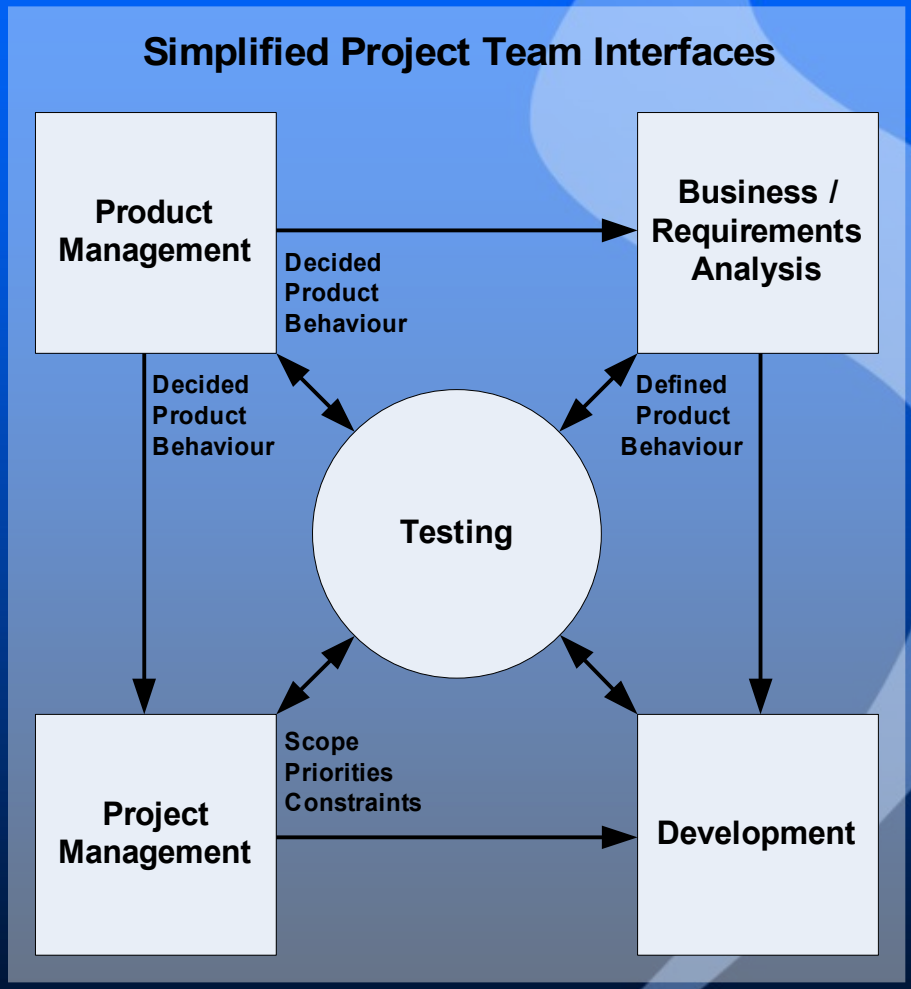
Resolution Implications

Silverpath Technologies

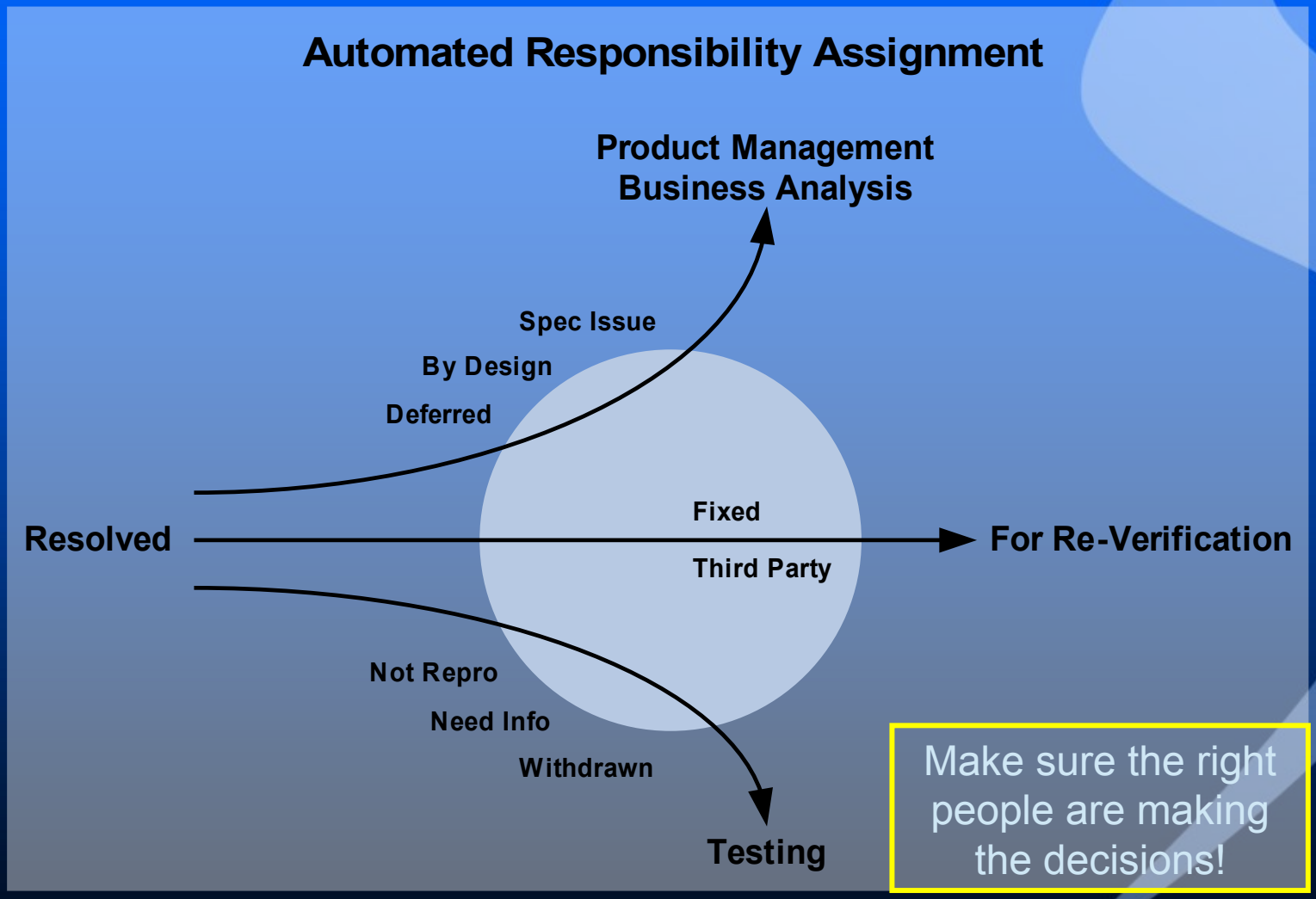
- Each resolution option should make sure the right questions are asked of the right people:
  - Who decides how the issue is to be truly resolved?
  - Who is assigned the issue next?
- Resolution data can also be leveraged to make broader observations & conclusions from trends, eg:
  - Do the testers need more testing or product training?
  - Are the requirements or design poorly captured?
  - Is technical debt becoming an issue?
- Which resolution options would help with each?

Consider Roles on the Project Team

- **Product Management**: must decide product behaviour & priorities
- **Project Management**: negotiates scope & priorities vs. constraints (schedule, resources, etc)
- **Business Analysis**: describes the product behaviour to be implemented
- **Development**: implements the described product behaviour
- **Testing**: verifies the implemented product behaviour against what was described; connecting the other roles with information



Process Automation using Resolution



Common Defect Report Attributes

Silverpath Technologies

- Resolution is just one of many typical fields or attributes:
  - Status
  - Assigned To
  - Priority
  - Severity
  - Functional Area
  - Feature
  - How Found
  - Type
  - Environment
  - ***Resolution***
  - Opened Version
  - Opened By
  - Opened Date
  - Related Test Case(s) or Requirement(s)
  - History or Audit Trail
- What other opportunities for efficiency can you find?

Defect Lifecycle Optimization Tips

- Defect management needs thoughtful consideration to ensure that communication & turnaround time is as efficient and collaborative as possible
- Without specific tracking of defect resolutions, the true defect find rate & defect clustering in the code is obscured:
  - By Duplicates, Not Repros, By Designs, Enhancements, and Feature Requests
- An effective defect lifecycle ensures that:
  - The highest ratio of valid & unique defects are being reported
  - Total time required to address each defect is minimized
  - The right role in the project team is making the decision for each next step in the defect lifecycle
- Improved definition & analysis of the data captured can drive improvements in processes and training, resulting in more successful projects!

Thinking Through Testing

Silverpath Technologies

For our latest updates:

- Visit ThinkTesting.com
- Follow @ThinkTesting

Discussing "right-fit" approaches for software testing

We are always sharing our ideas on crafting “right-fit” approaches to software testing. We are sure you will find something you can apply to your own projects and organizational environment.

Thinking Through Testing