DEPARTMENT OF COMPUTER & INFORMATION SCIENCES

University of Strathclyde Science

Software Engineering

Lecture 8: Maintenance

Billy Wallace
w.wallace@strath.ac.uk

CS993

Lecture Outline

- When maintenance is different from developing new code.
- Issue tracking systems.
- Technical debt.
- Code smells.
- Refactoring.

What is maintenance?

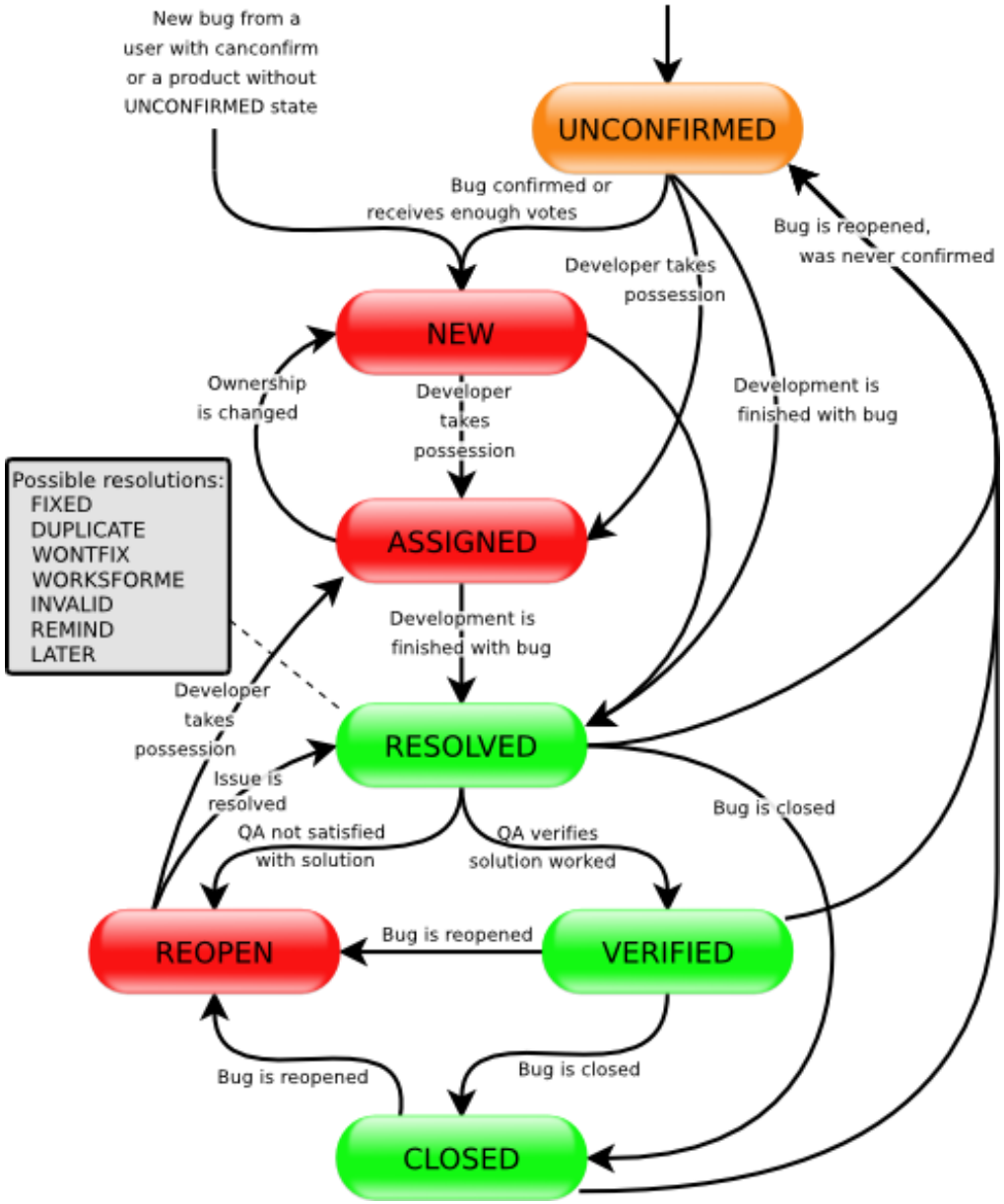
- Adding a feature.
- Fixing a bug.
- Improving the design.
- Optimising resource usage.

Maintenance & Support

- It is common with “big ticket” software to include maintenance & support as a cost.
- This can be as much as 25% of the cost of the software annually.
- For this, you give help on the software and supply new versions as they become available.
- This will include a service level agreement. 24x7 support will be more expensive than weekday, working hours support.
- Customers need to list the severity of the problem when they report it, this will have an impact on how quickly you need to respond.

Bug Tracking Systems

- “Tickets” will be raised regarding issues. These are called various things, including Problem Reports (PRs).
- Support requests will be treated differently than those raised internally as we will need to meet our SLA for a support request.
- Tools include: Bugzilla, Jira and commercial tools like YouTrack.
- A workflow will structure how we work on reported issues.


New bug from a user with canconfirm or a product without UNCONFIRMED state
UNCONFIRMED
Bug confirmed or receives enough votes
Bug is reopened, was never confirmed
Developer takes possession
NEW
Ownership is changed
Developer takes possession
Development is finished with bug
Possible resolutions:
FIXED
DUPLICATE
WONTFIX
WORKSFORME
INVALID
REMIND
LATER
ASSIGNED
Development is finished with bug
Developer takes possession
RESOLVED
Issue is resolved
Bug is closed
QA not satisfied with solution
QA verifies solution worked
REOPEN
Bug is reopened
VERIFIED
Bug is reopened
Bug is closed
CLOSED

Same old, same old

- New features and bug fixes follow essentially the same process as developing a new system.
- Prioritise and add them to the next sprint.
- Optimising resources is similar. Remember to use a profiler to track down the issue.

Except …

- However, if these have been raised because of support requests, then you may have an SLA that causes you to make a change and ship it quickly.
- A different team might be responsible for this so that the development team can continue working to a carefully planned schedule.
- Ultimately you might need the dev team to look at particularly horrible problems, but first-line support should be there to try to keep them on schedule.
- If you now have different customers with different releases, you must track what is deployed where. Otherwise you can't diagnose and fix the next problem from the same customer.
- This can also cause problems with upgrades.

Improving the Design

- If it's not broke, don't fix it …
- … unless you want to pay-down your "Technical Debt".
- Refactoring is a structured way to do this.
- Code smells are a way to spot things to refactor.

University of Strathclyde Science

DEPARTMENT OF COMPUTER & INFORMATION SCIENCES

Code Smells

- Duplicate code
- Long method
- Large class
- Long parameter list
- Switch statements
- Comments

- Look at Chapter 3 in Fowler's Refactoring book.

Desirable properties of classes

- **Cohesion** refers to how closely all the routines in a class or all the code in a routine support a central purpose (Steve McConnell)
- **Weak Coupling**: If two modules communicate, they should exchange as little information as possible (Bertrand Meyer)
- **Small size** - big things are complicated (Billy)

Refactoring

- These design changes are done without changing the behaviour of the system.
- Do not refactor and change functionality at the same time.
- You need good test case coverage to do this properly.
- You can do refactoring manually, but the IDEs are a Godsend.
- If you learn nothing else, learn to use the rename refactoring in Eclipse.

Refactoring

- Rename method.
- Extract method.
- Extract interface.
- Pull up field.

Summary

- Mostly we are doing what we did before, but SLAs can mean changing our procedures to cope.
- Refactoring is something new we can do to improve the design.
- Consider that old software dies and we need to consider how we EOL it properly.

University of Strathclyde Science

Moving on from here …

- In the practical this week, we'll look at refactoring.
- Do some reading on code smells and refactoring.
- Try out refactoring in Eclipse (or another IDE).

University of Strathclyde Glasgow