

Agile Maintenance

Jason Gorman

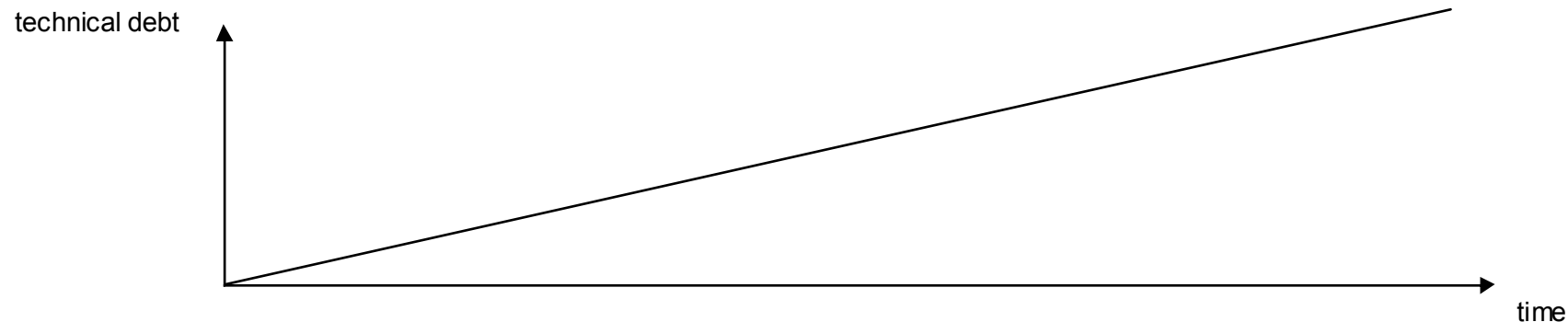
Technical Debt & Productivity



rowing



sinking



Technical Debt & Maintenance



bailing

technical debt



staying afloat!

time

What Is Technical Debt?

- Anything that diverts people from rowing to bailing (or sinking!)
 - Bugs
 - Failure to meet functional requirements
 - Failure to meet non-functional requirements
 - Poor design quality
 - Code that's harder to modify or add new features to
 - Badly designed user interfaces that require more intervention to support users

What Is Maintenance?

- Improving Quality of Existing Code
 - Fixing Bugs
 - Maintaining or Improving Design Quality
 - Improving Usability
 - Improving Performance & Scalability
 - Improving Security

Maintenance Is...

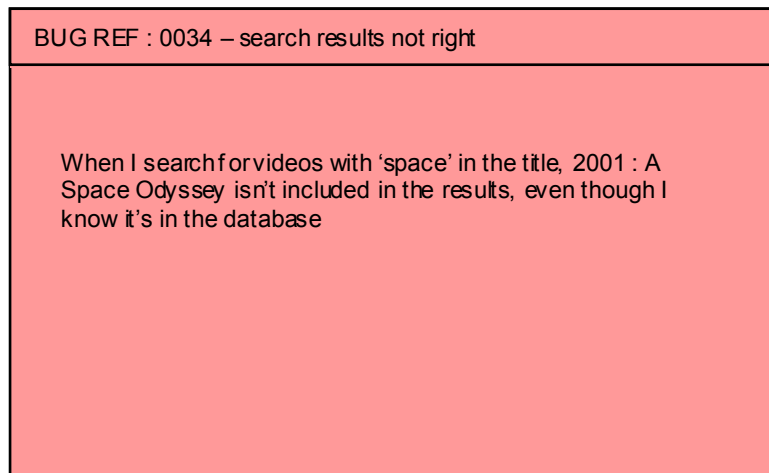


Bailing
- *Paying off the technical debt*

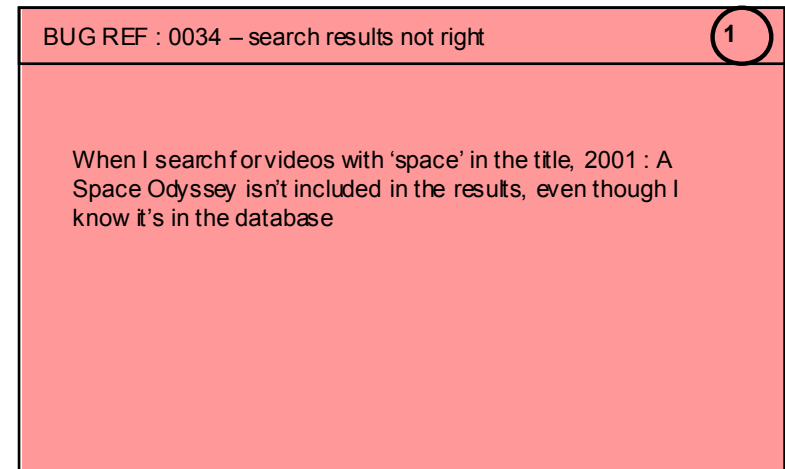


Fixing the leaks
- *Preventing re-introduction of problems*

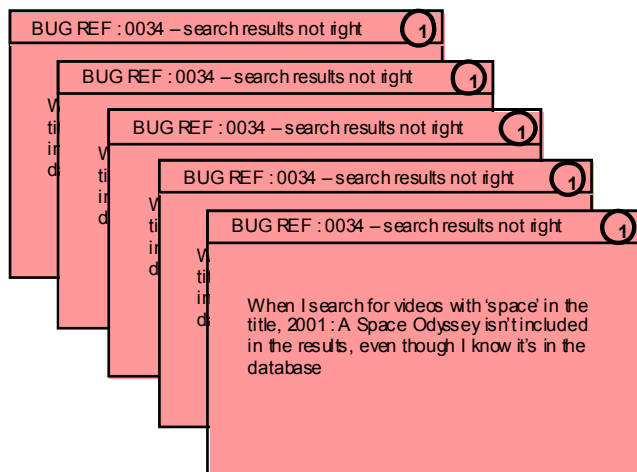
Agile Bug Fixing - Scheduling



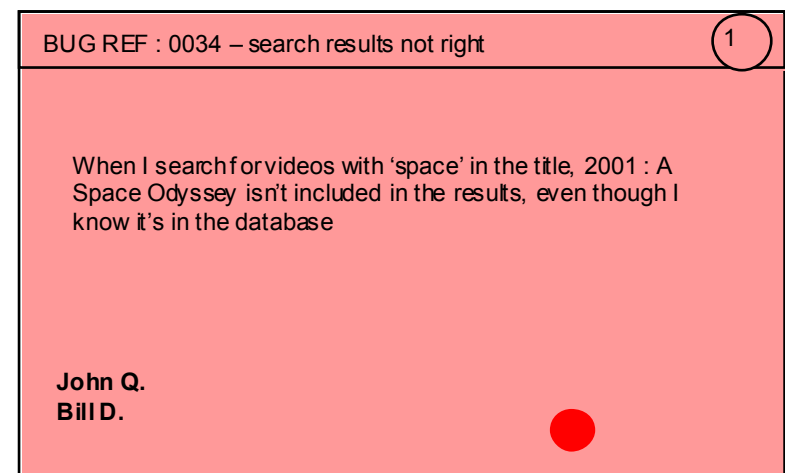
1. Bug is reported



2. Developer estimates relative effort required to fix

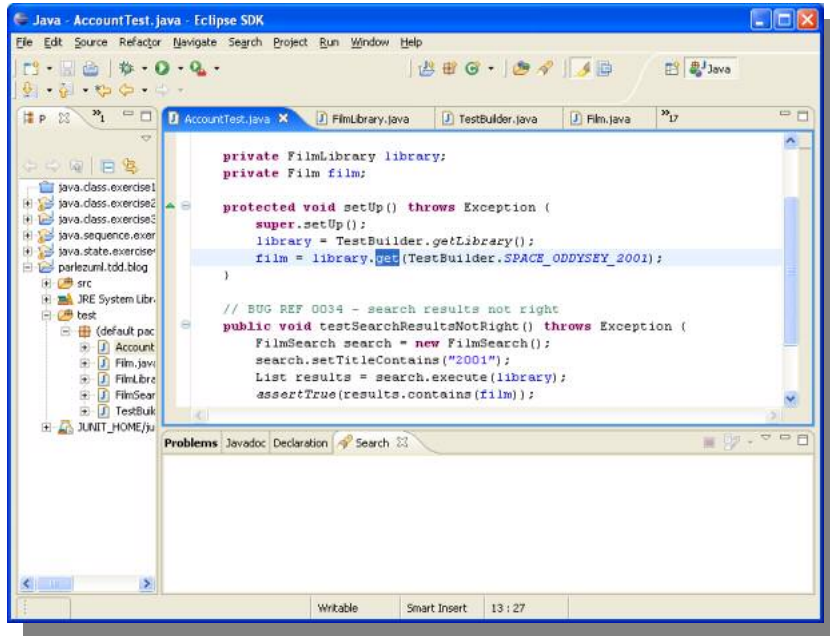


3. "Customer" schedules bug fixes in iteration

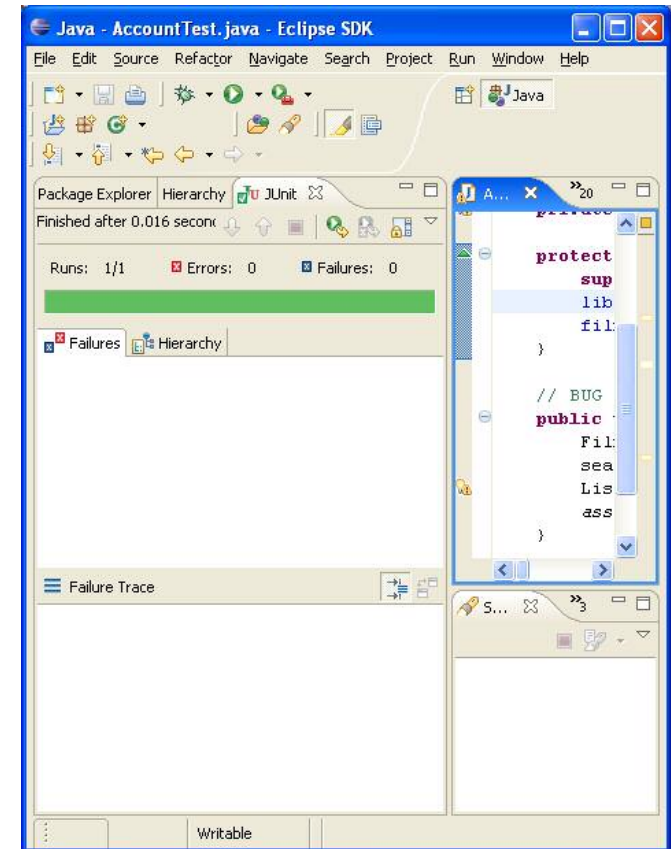
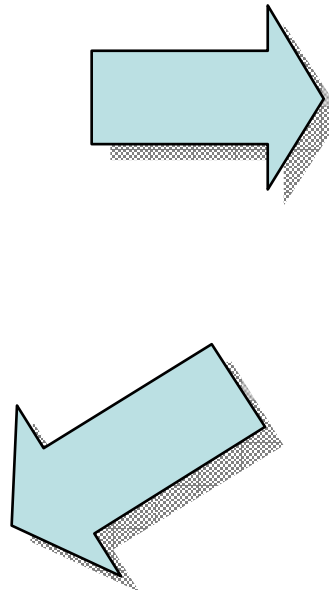


4. Developers start with the first bug scheduled & agree **acceptance tests** with customer

Agile Bug Fixing - Executing



5. Write a test that reproduces the bug



6. Fix The Bug (by passing the test)

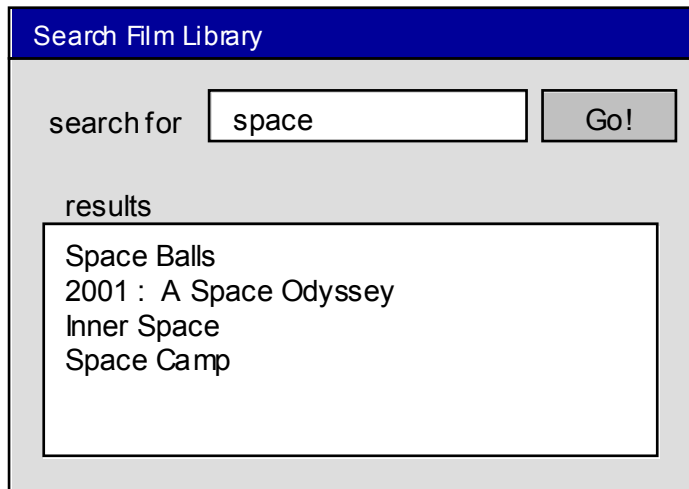
BUG REF : 0034 – search results not right 1

When I search for videos with 'space' in the title, 2001 : A Space Odyssey isn't included in the results, even though I know it's in the database

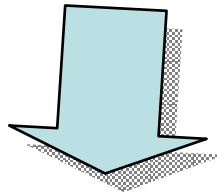
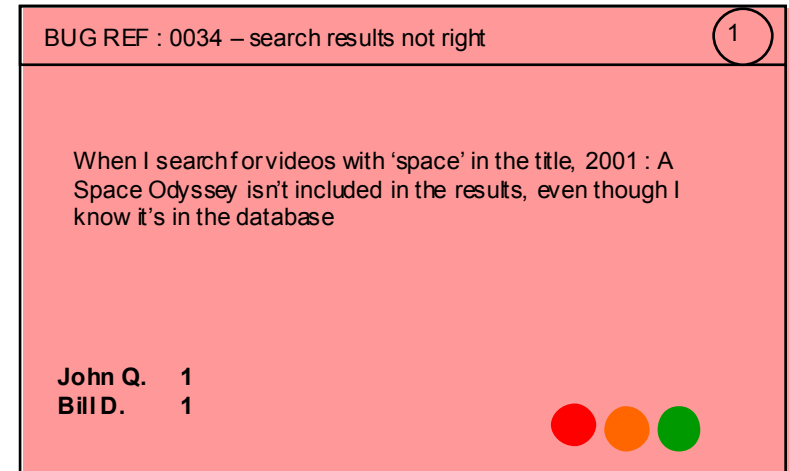
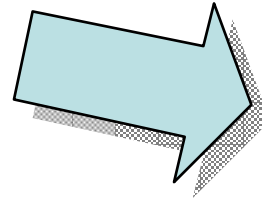
John Q. 1
Bill D. 1

7. Schedule bug fix for acceptance testing (and record actual effort)

Agile Bug Fixing – Acceptance Testing

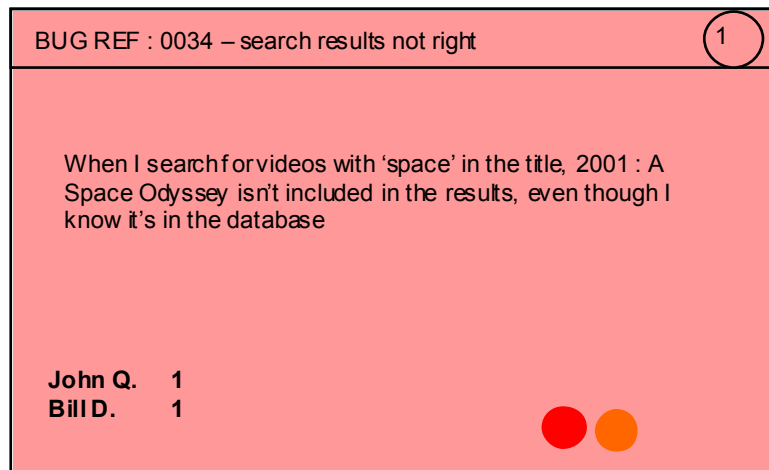


8. "Customer" executes acceptance test script




9b. If it fails, pass bug back to developer to fix again

9a. If it passes the test, bug fix is complete – add estimate to iterations velocity

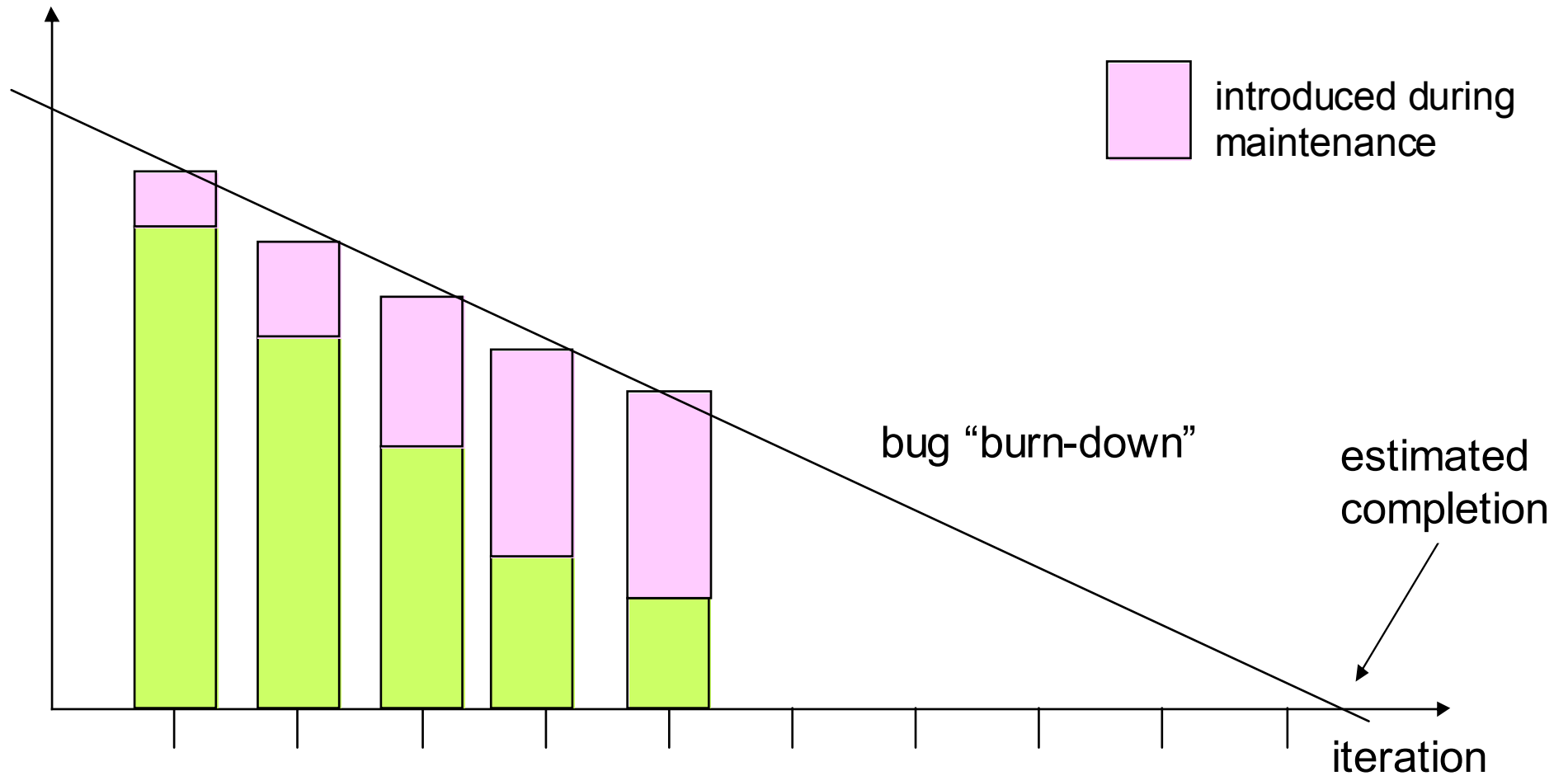


Agile Bug Fixing - Tracking

total outstanding bugs (I.E.D)

 scheduled for maintenance release

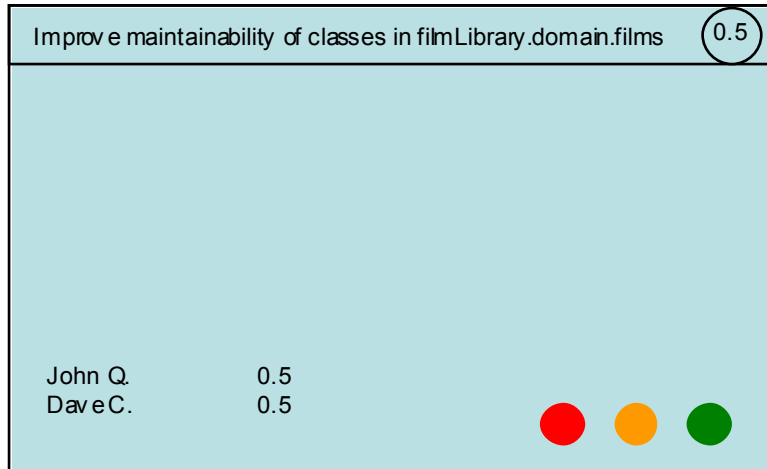
 introduced during maintenance



Bug Patterns

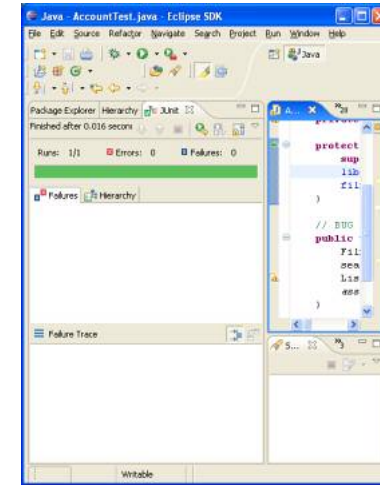
- Developers tend to make the same kind of mistakes more than once
- Group similar bugs together into defect classifications
 - E.g., referencing features on null objects
- Share knowledge on how to avoid these bugs in future
 - What tests show them up?
 - Weekly bug reviews
 - Pair programming
 - A “most wanted” poster on every desktop!

Design Quality & Refactoring

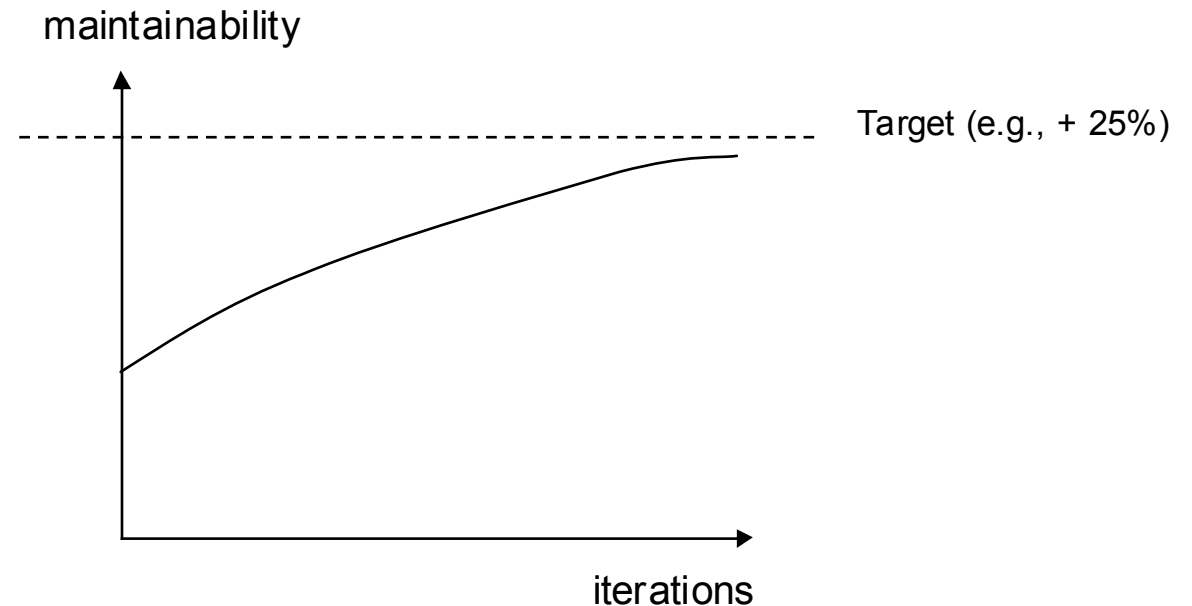


Significant refactorings are scheduled alongside bug fixes

Automated tests are used to ensure refactoring hasn't broken anything



Progress can be tracked using design quality metrics against targets



Refactoring Legacy Code

- If the code being refactored has no unit tests, write tests for the *context* of that refactoring
 - i.e., tests for the function that must behave the same after the refactoring (“black box tests”)
- Consider the value of refactoring vs. the cost
 - *Prioritise* refactorings