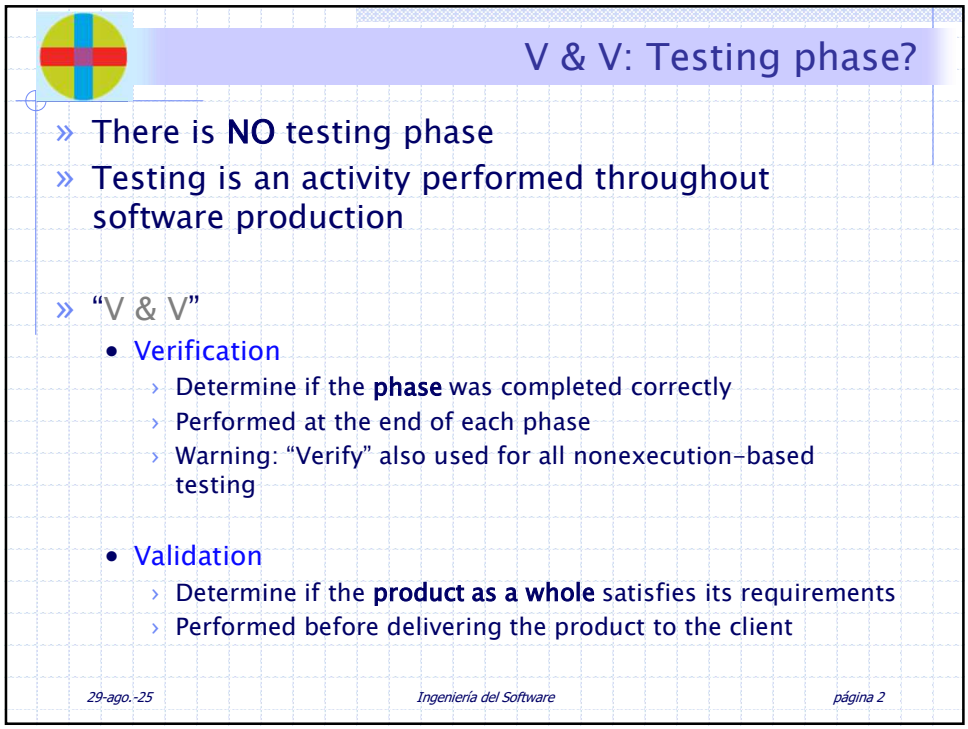
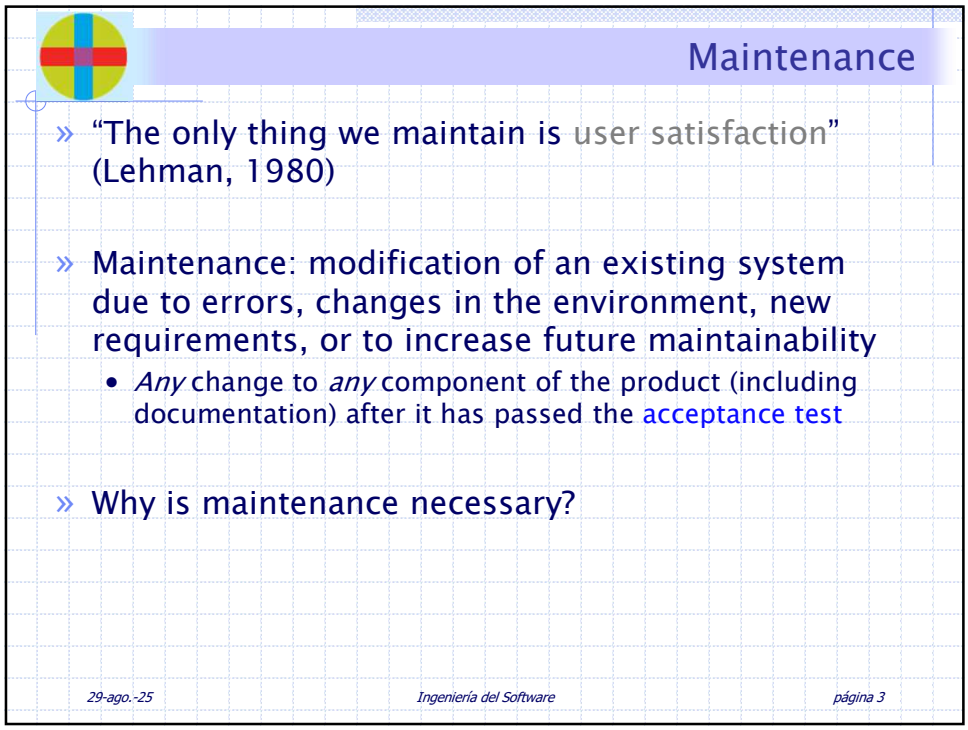

Verificación, validación y mantenimiento
Ingeniería del Software
Curso 2025/2026
Universidad San Pablo-CEU
Escuela Politécnica Superior
Campus de Montepríncipe

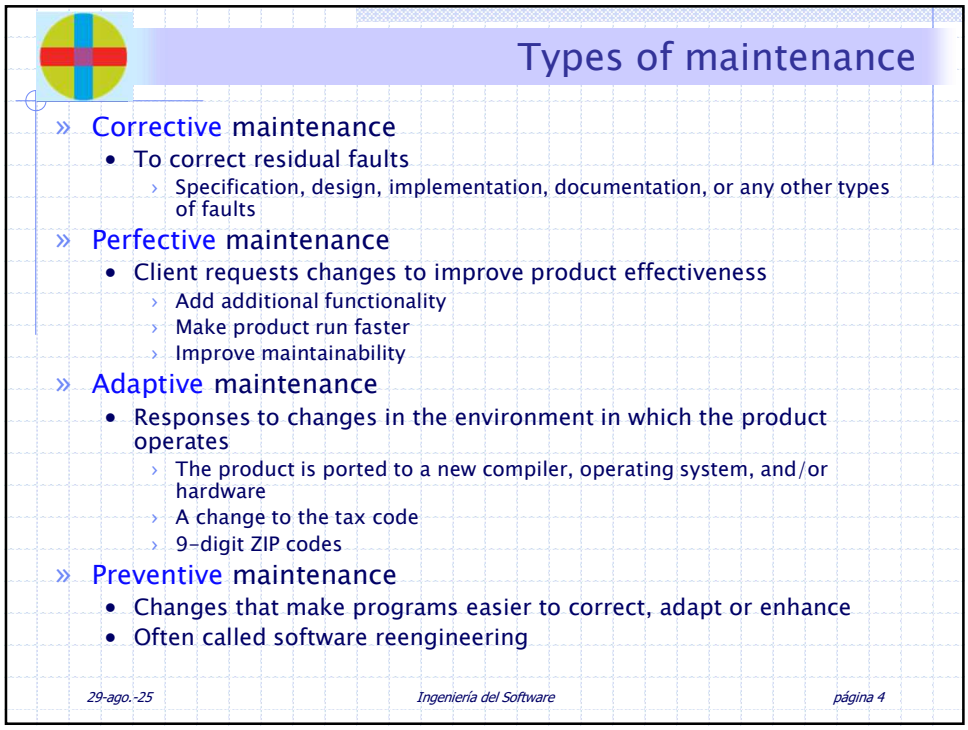
1


V & V: Testing phase?
There is NO testing phase
Testing is an activity performed throughout software production
"V & V"
Verification
Determine if the phase was completed correctly
Performed at the end of each phase
Warning: "Verify" also used for all nonexecution-based testing
Validation
Determine if the product as a whole satisfies its requirements
Performed before delivering the product to the client
29-ago.-25
Ingeniería del Software
página 2

2

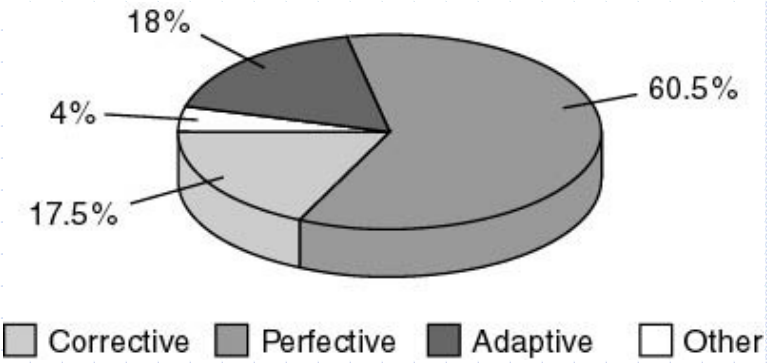

Maintenance
» "The only thing we maintain is user satisfaction" (Lehman, 1980)
» Maintenance: modification of an existing system due to errors, changes in the environment, new requirements, or to increase future maintainability
• Any change to any component of the product (including documentation) after it has passed the acceptance test
» Why is maintenance necessary?
29-ago.-25 Ingeniería del Software página 3

3


Types of maintenance
» Corrective maintenance
• To correct residual faults
› Specification, design, implementation, documentation, or any other types of faults
» Perfective maintenance
• Client requests changes to improve product effectiveness
› Add additional functionality
› Make product run faster
› Improve maintainability
» Adaptive maintenance
• Responses to changes in the environment in which the product operates
› The product is ported to a new compiler, operating system, and/or hardware
› A change to the tax code
› 9-digit ZIP codes
» Preventive maintenance
• Changes that make programs easier to correct, adapt or enhance
• Often called software reengineering
29-ago.-25 Ingeniería del Software página 4

4

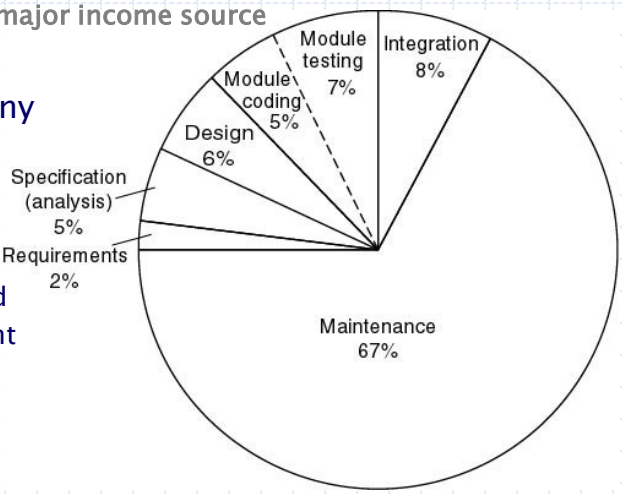
Distribution of types of maintenance



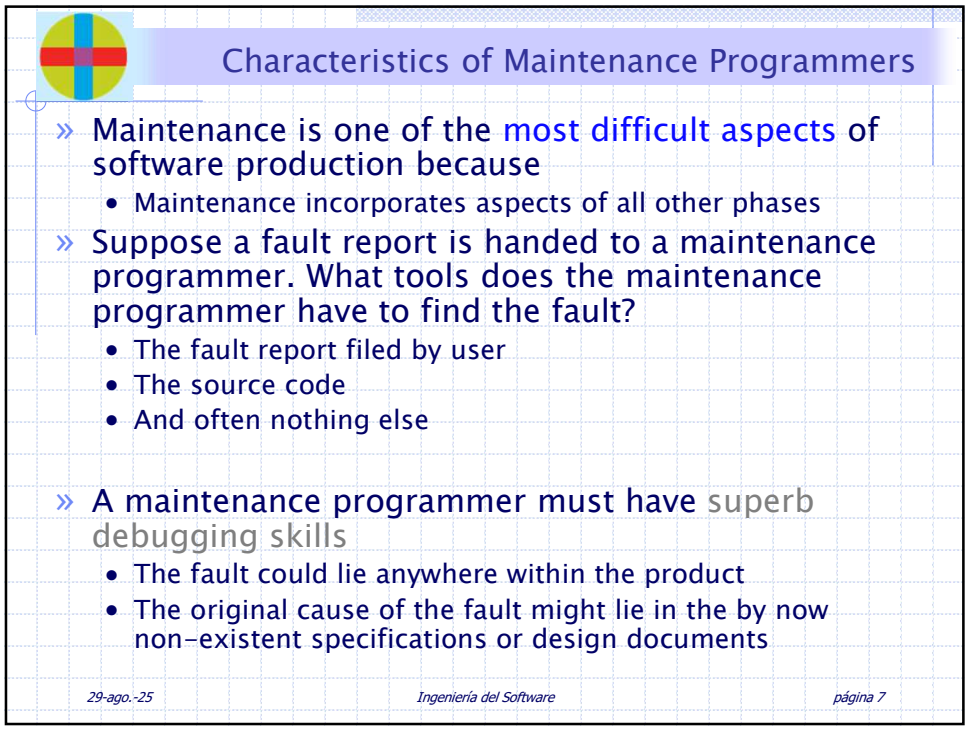
5

Difficulty of maintenance

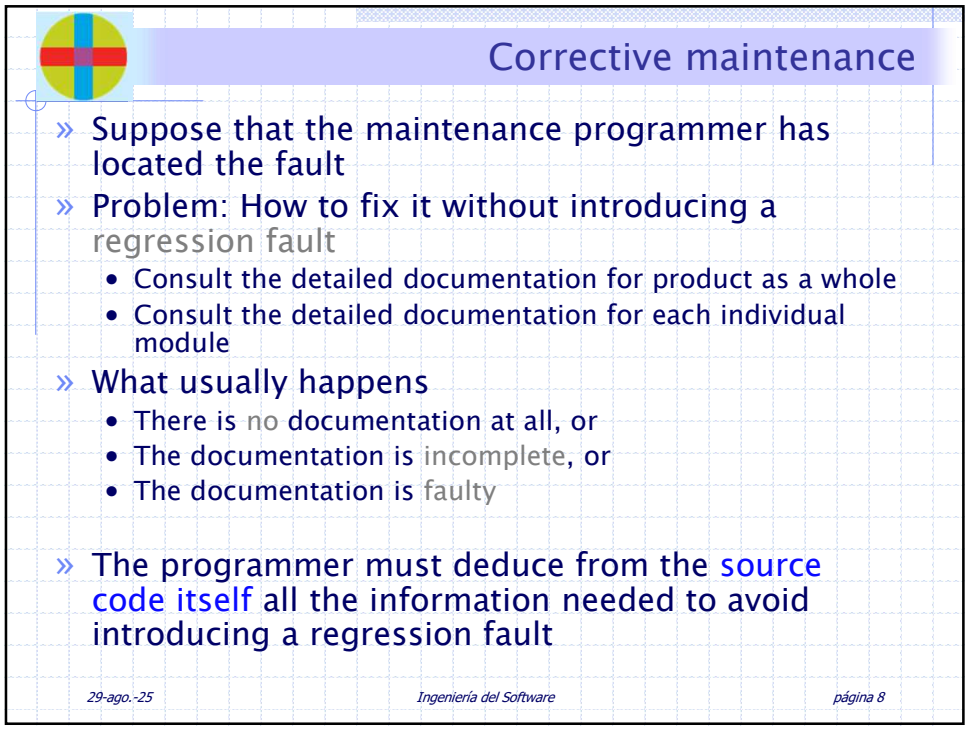
- About 67% of the total cost of a product is obtained during the maintenance phase
  Maintenance is a major income source
- Nevertheless, even today many organizations assign it to:
  - Unsupervised beginners, and
  - Less competent programmers
- Why?



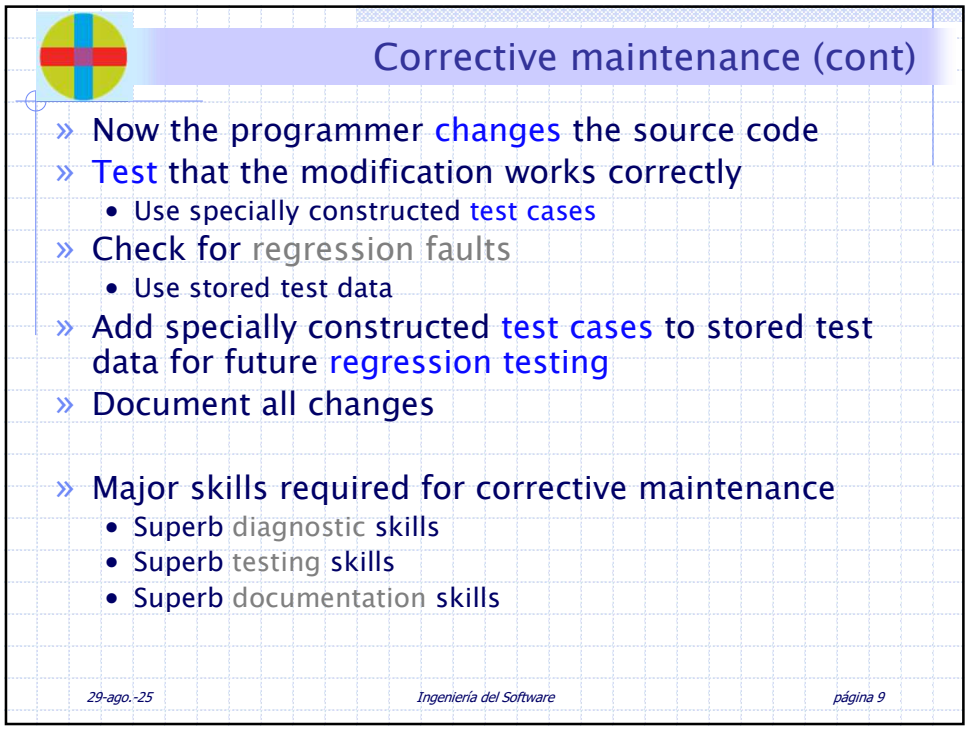
6


Characteristics of Maintenance Programmers
» Maintenance is one of the most difficult aspects of software production because
• Maintenance incorporates aspects of all other phases
» Suppose a fault report is handed to a maintenance programmer. What tools does the maintenance programmer have to find the fault?
• The fault report filed by user
• The source code
• And often nothing else
» A maintenance programmer must have superb debugging skills
• The fault could lie anywhere within the product
• The original cause of the fault might lie in the by now non-existent specifications or design documents
29-ago.-25 Ingeniería del Software página 7

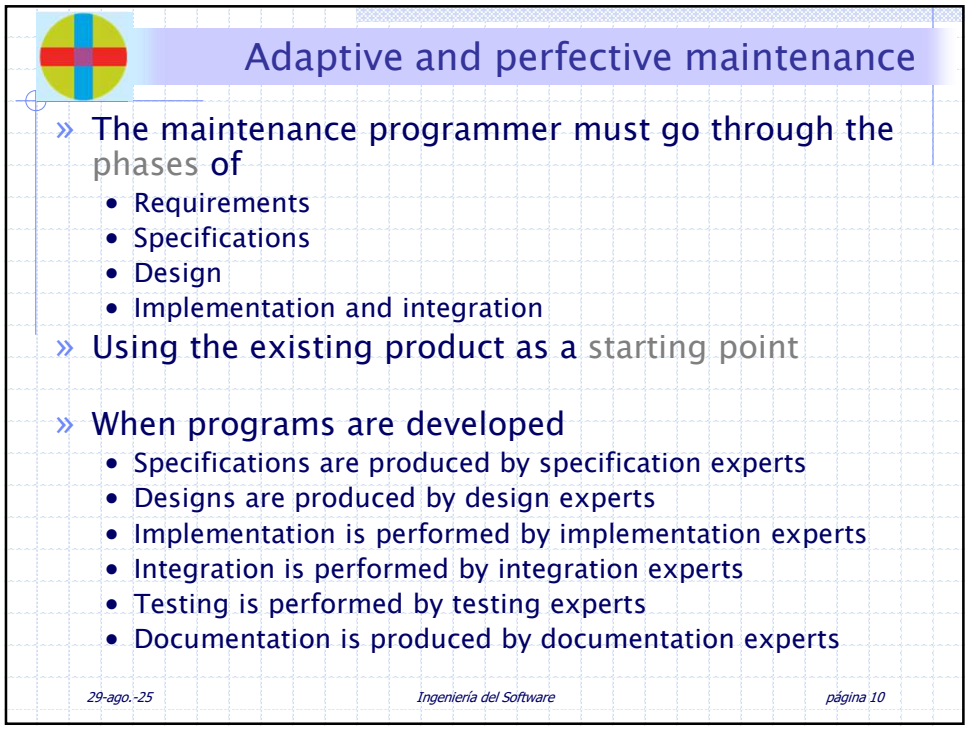
7


Corrective maintenance
» Suppose that the maintenance programmer has located the fault
» Problem: How to fix it without introducing a regression fault
• Consult the detailed documentation for product as a whole
• Consult the detailed documentation for each individual module
» What usually happens
• There is no documentation at all, or
• The documentation is incomplete, or
• The documentation is faulty
» The programmer must deduce from the source code itself all the information needed to avoid introducing a regression fault
29-ago.-25 Ingeniería del Software página 8

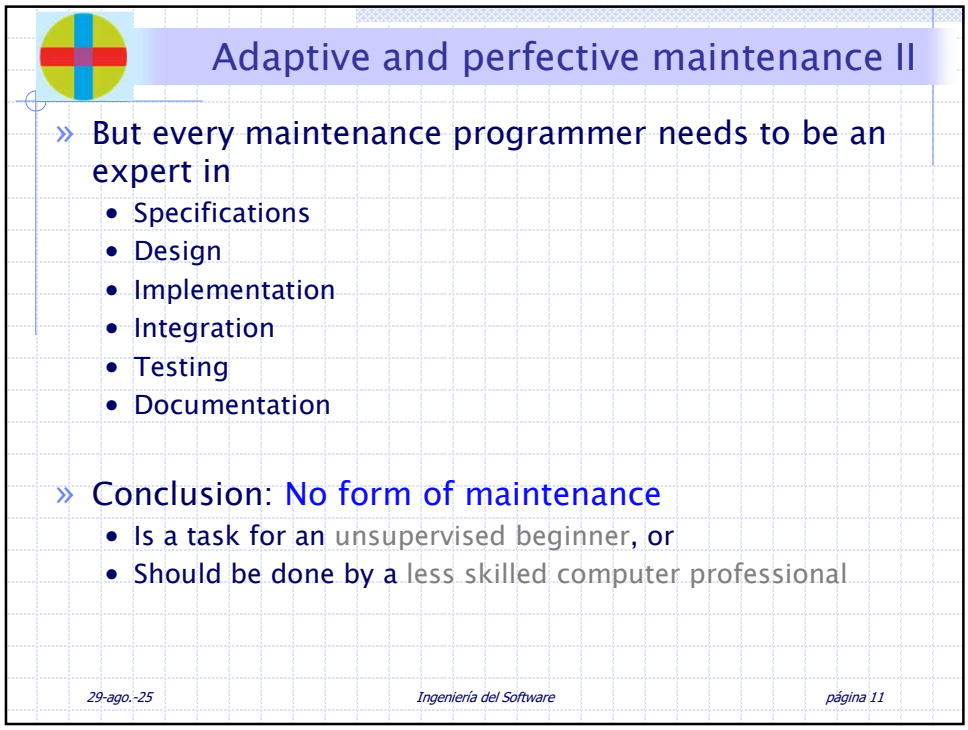
8


Corrective maintenance (cont)
» Now the programmer changes the source code
» Test that the modification works correctly
• Use specially constructed test cases
» Check for regression faults
• Use stored test data
» Add specially constructed test cases to stored test data for future regression testing
» Document all changes
» Major skills required for corrective maintenance
• Superb diagnostic skills
• Superb testing skills
• Superb documentation skills
29-ago.-25
Ingeniería del Software
página 9

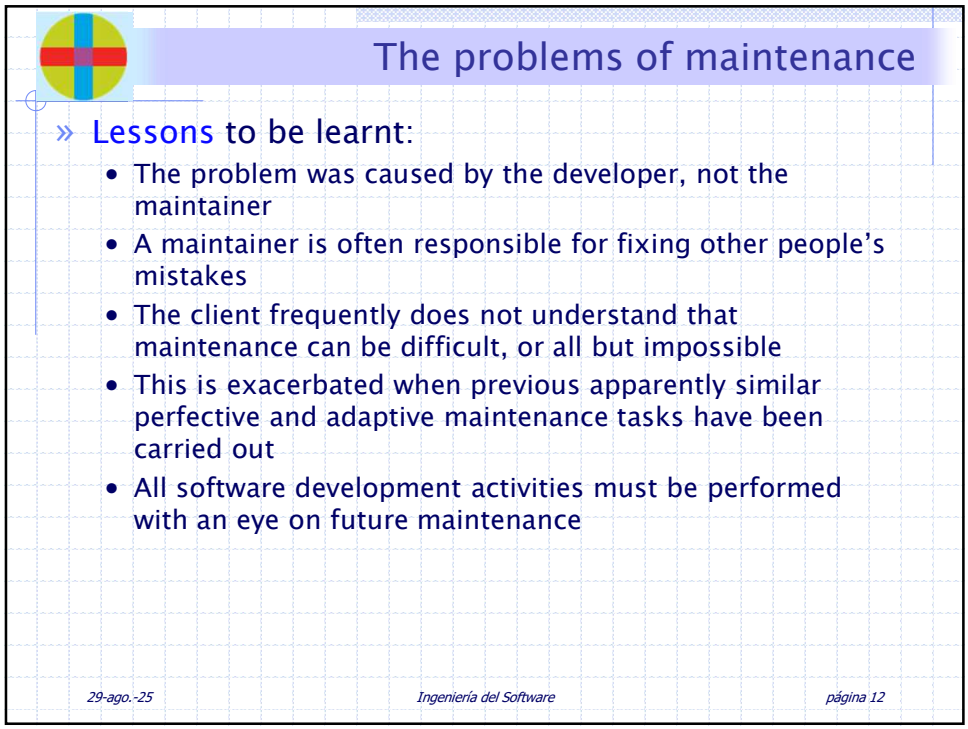
9


Adaptive and perfective maintenance
» The maintenance programmer must go through the phases of
• Requirements
• Specifications
• Design
• Implementation and integration
» Using the existing product as a starting point
» When programs are developed
• Specifications are produced by specification experts
• Designs are produced by design experts
• Implementation is performed by implementation experts
• Integration is performed by integration experts
• Testing is performed by testing experts
• Documentation is produced by documentation experts
29-ago.-25
Ingeniería del Software
página 10

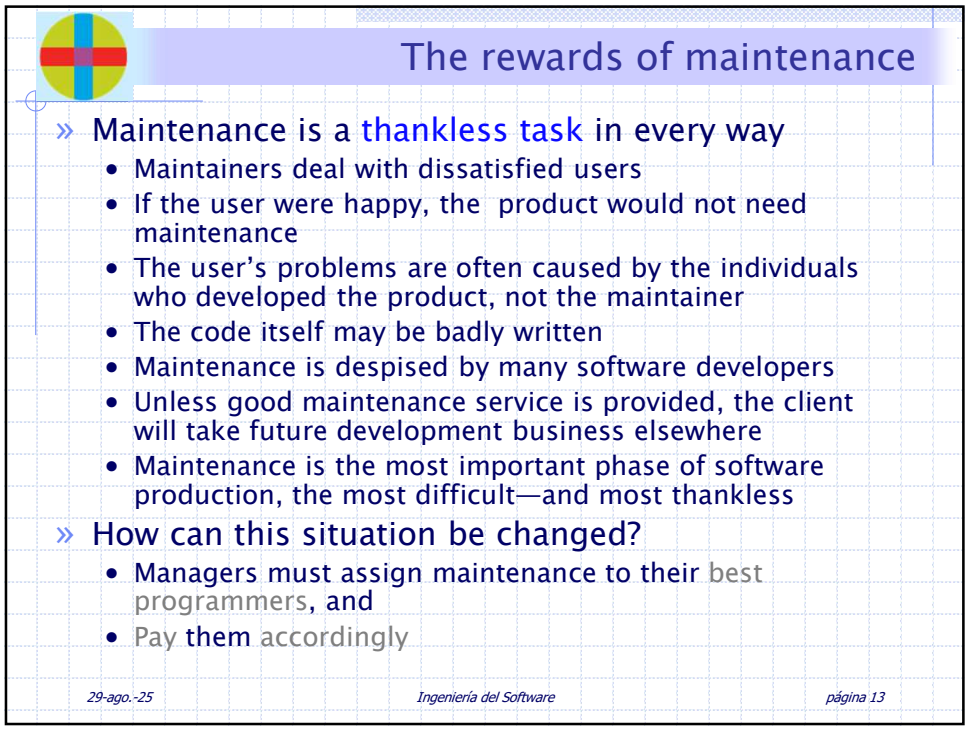
10


Adaptive and perfective maintenance II
» But every maintenance programmer needs to be an expert in
• Specifications
• Design
• Implementation
• Integration
• Testing
• Documentation
» Conclusion: No form of maintenance
• Is a task for an unsupervised beginner, or
• Should be done by a less skilled computer professional
29-ago.-25 Ingeniería del Software página 11

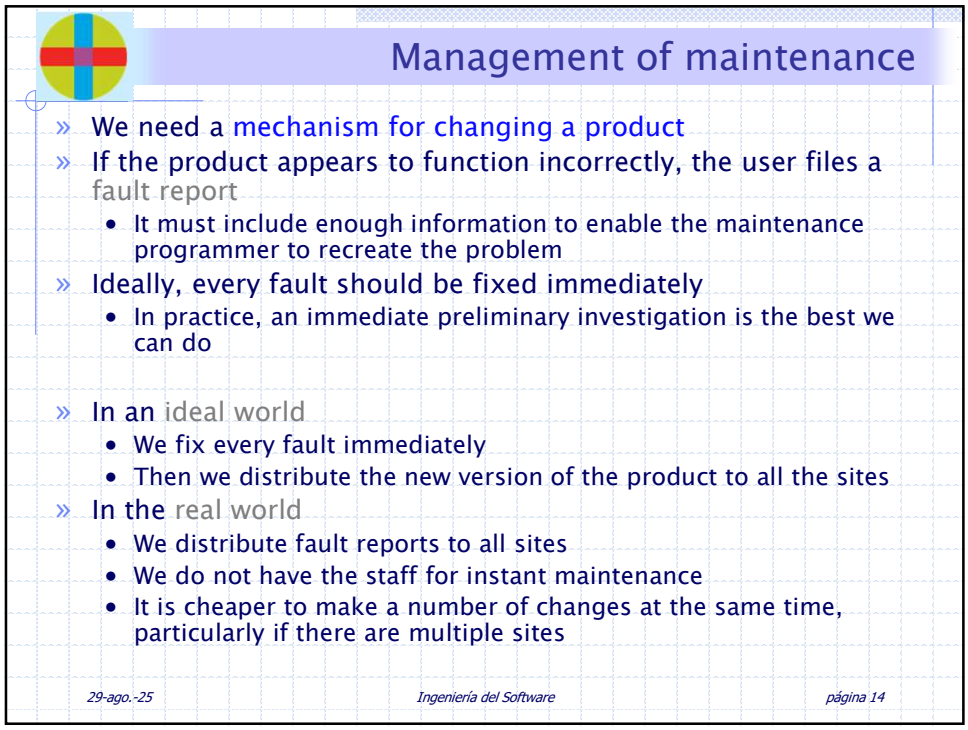
11


The problems of maintenance
» Lessons to be learnt:
• The problem was caused by the developer, not the maintainer
• A maintainer is often responsible for fixing other people's mistakes
• The client frequently does not understand that maintenance can be difficult, or all but impossible
• This is exacerbated when previous apparently similar perfective and adaptive maintenance tasks have been carried out
• All software development activities must be performed with an eye on future maintenance
29-ago.-25 Ingeniería del Software página 12

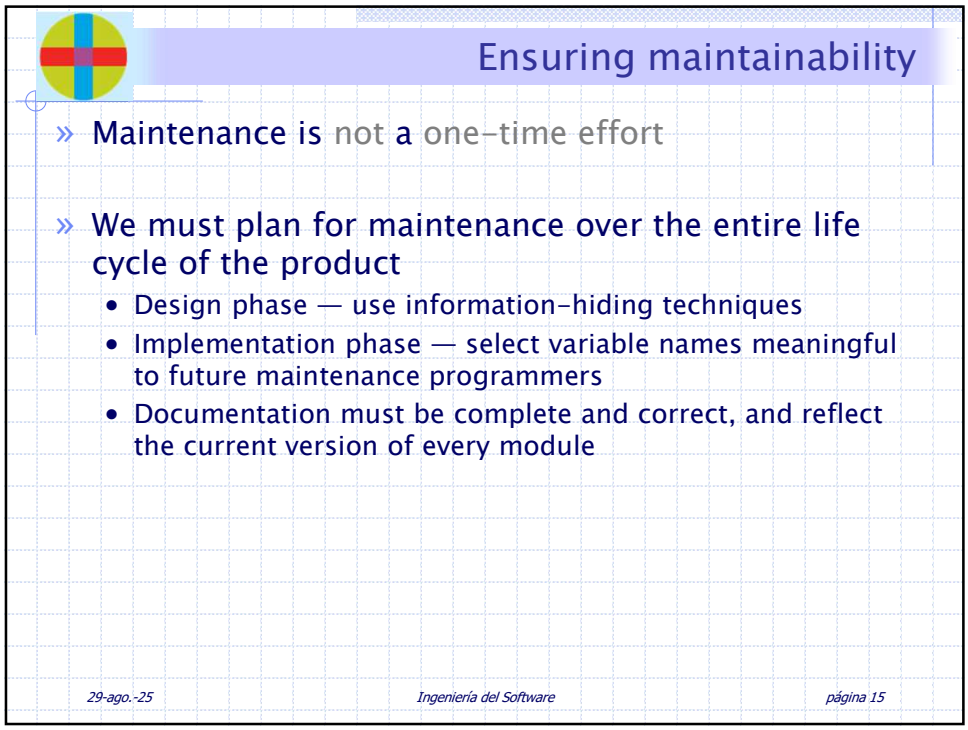
12


The rewards of maintenance
» Maintenance is a thankless task in every way
• Maintainers deal with dissatisfied users
• If the user were happy, the product would not need maintenance
• The user's problems are often caused by the individuals who developed the product, not the maintainer
• The code itself may be badly written
• Maintenance is despised by many software developers
• Unless good maintenance service is provided, the client will take future development business elsewhere
• Maintenance is the most important phase of software production, the most difficult—and most thankless
» How can this situation be changed?
• Managers must assign maintenance to their best programmers, and
• Pay them accordingly
29-ago.-25 Ingeniería del Software página 13

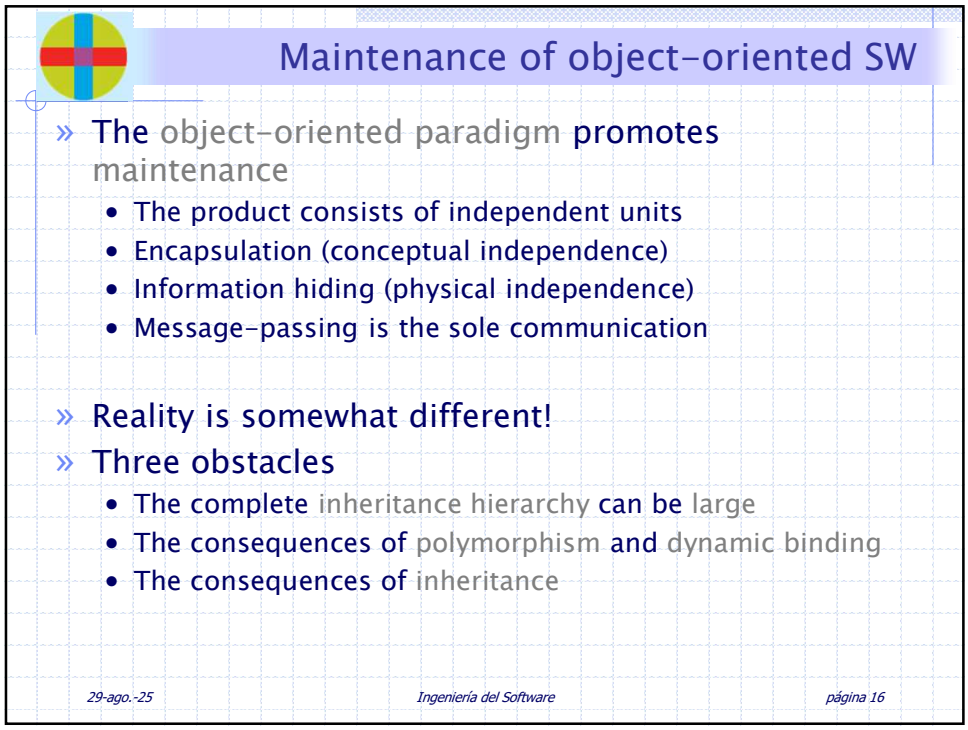
13


Management of maintenance
» We need a mechanism for changing a product
» If the product appears to function incorrectly, the user files a fault report
• It must include enough information to enable the maintenance programmer to recreate the problem
» Ideally, every fault should be fixed immediately
• In practice, an immediate preliminary investigation is the best we can do
» In an ideal world
• We fix every fault immediately
• Then we distribute the new version of the product to all the sites
» In the real world
• We distribute fault reports to all sites
• We do not have the staff for instant maintenance
• It is cheaper to make a number of changes at the same time, particularly if there are multiple sites
29-ago.-25 Ingeniería del Software página 14

14


Ensuring maintainability
» Maintenance is not a one-time effort
» We must plan for maintenance over the entire life cycle of the product
• Design phase — use information-hiding techniques
• Implementation phase — select variable names meaningful to future maintenance programmers
• Documentation must be complete and correct, and reflect the current version of every module
29-ago.-25
Ingeniería del Software
página 15

15


Maintenance of object-oriented SW
» The object-oriented paradigm promotes maintenance
• The product consists of independent units
• Encapsulation (conceptual independence)
• Information hiding (physical independence)
• Message-passing is the sole communication
» Reality is somewhat different!
» Three obstacles
• The complete inheritance hierarchy can be large
• The consequences of polymorphism and dynamic binding
• The consequences of inheritance
29-ago.-25
Ingeniería del Software
página 16

16

Retirement

- Good software is maintained
- Sometimes software is rewritten from scratch
  - Software is now unmaintainable because
    - A drastic change in design has occurred
    - The product must be implemented on a totally new hardware/operating system
    - Documentation is missing or inaccurate
    - Hardware is to be changed—it may be cheaper to rewrite the software from scratch than to modify it
- True retirement is a rare event

17

¿Preguntas?

18