Managing Software Quality

Main issues:

- Quality cannot be added as an afterthought
- To measure is to know
- Product quality vs process quality

Commitment to quality pays off

Approaches to quality

- Quality of the product versus quality of the process
- Check whether (product or process) *conforms to* certain norms
- Improve quality by improving the product or process

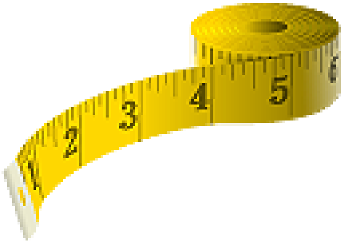
Approaches to quality

	Conformance	Improvement
Product	ISO 9126	‘best practices’
Process	ISO 9001 SQA	CMM SPICE Bootstap

What is quality?

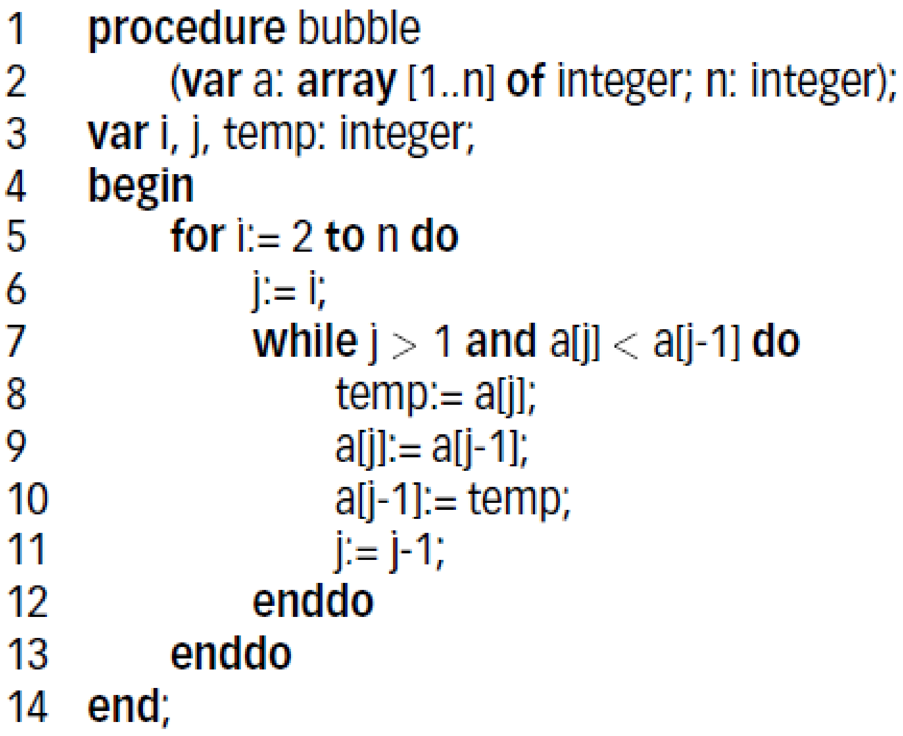
software

+

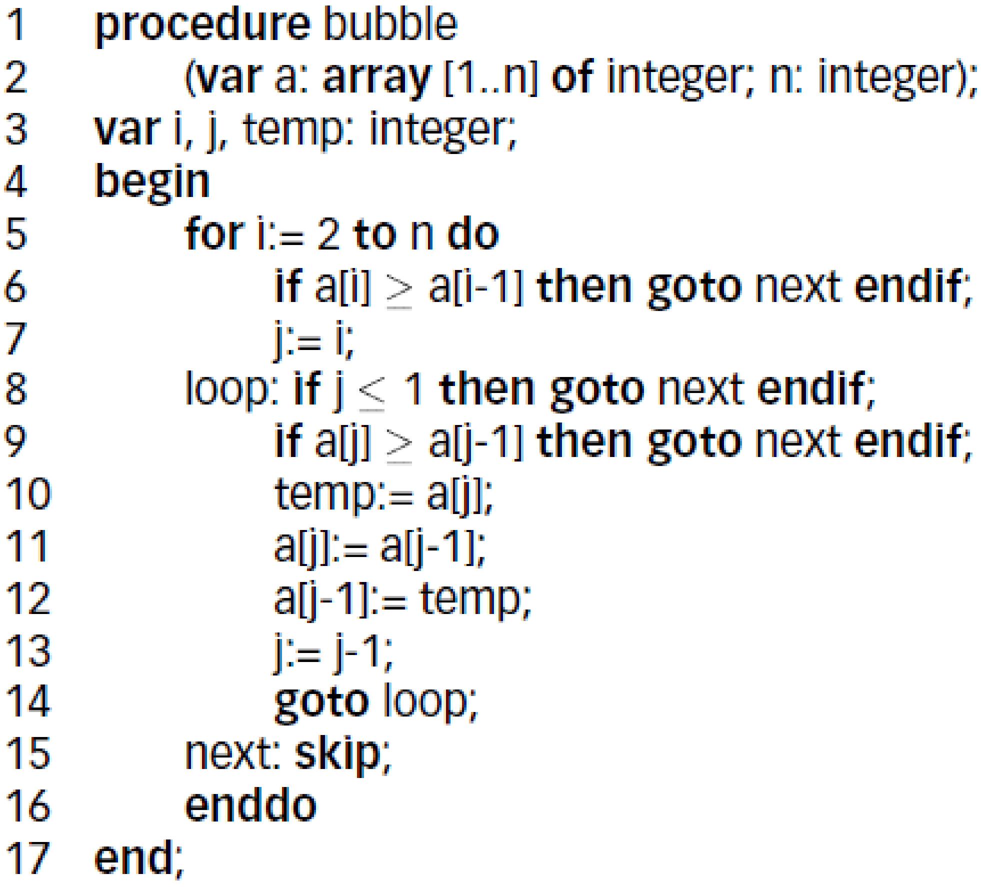


measures

Complexity



```
procedure bubble
    (var a: array [1..n] of integer; n: integer);
var i, j, temp: integer;
begin
    for i:= 2 to n do
        j:= i;
        while j > 1 and a[j] < a[j-1] do
            temp:= a[j];
            a[j]:= a[j-1];
            a[j-1]:= temp;
            j:= j-1;
        enddo
    enddo
end;
```



```
procedure bubble
    (var a: array [1..n] of integer; n: integer);
var i, j, temp: integer;
begin
    for i:= 2 to n do
        if a[i] ≥ a[i-1] then goto next endif;
        j:= i;
    loop: if j ≤ 1 then goto next endif;
        if a[j] ≥ a[j-1] then goto next endif;
        temp:= a[j];
        a[j]:= a[j-1];
        a[j-1]:= temp;
        j:= j-1;
        goto loop;
    next: skip;
    enddo
end;
```

Measures and Numbers

- **Complexity of a Program in single numeric value**
- **Larger values for more complex programs**
- **If P1 is more complex than P2, then**
  - C(P1) > C(P2)
- **C is the complexity mapping**
- **Use: e.g Could be used for planning maintenance**

How to measure “complexity”?

- The length of the program?
- The number of goto’s?
- The number of if-statements?
- The sum of these numbers?
- Yet something else?

Scale types

- **Nominal: just classification**
  - e.g Color of eyes: Brown, Blue, Green
- **Ordinal: linear ordering (>)**
  - e.g this material is harder than this material
- **Interval: like ordinal, but interval between values is the same (so average has a meaning)**
  - Distance between successive value is same
  - E.g degree in farenheit
- **Ratio: like interval, but there is a 0 (zero) (so A can be twice B)**
  - Same as above emphasis on ‘zero value’
  - e.g temperatue in Kelvin
- **Absolute: counting number of occurrences**
  - E.g No of If statements in a program

Measures and Metrics

- **Measurement: is the mapping from empirical real world to the formal relational world**

- **Measure: is the number or symbol assigned to an attribute of an entity by this mapping.**

- **Metrics**
  - An attribute of an entity
  - The function which assigns value to that attribute
  - The unit in which this value is expressed and
  - Its scale type

Quality attributes (McCall)

- **Product operation**
  - Correctnessdoes it do what I want?
  - Reliability does it do it accurately all of the time?
  - Efficiency will it run on my hardware as well as it can?
  - Integrity is it secure?
  - Usability can I use it?
- **Product revision**
  - Maintainability can I fix it?
  - Testability can I test it?
  - Flexibility can I change it?
- **Product transition**
  - Portability will I be able to use it on another machine?
  - Reusability will I be able to reuse some of the software?
  - Interoperability will I be able to interface it with another system?

Taxonomy of quality attributes (ISO 9126)

- Functionality
- Reliability
- Usability
- Efficiency
- Maintainability
- Portability

ISO 9126 (cnt’d)

- **ISO 9126 measures ‘quality in use’: the extent to which users can achieve their goal**
- **Quality in use is modeled in four characteristics:**
  - Effectiveness
  - Productivity
  - Safety
  - Satisfaction

Perspectives on quality

- Transcendent (“I really like this program”)
- User-based (“fitness for use”)
- Product-based (based on attributes of the software)
- Manufacturing-based (conformance to specs)
- Value-based (balancing time and cost vs profits)

ISO 9001

- Model for quality assurance in design, development, production, installation and servicing

- Basic premise: confidence in product conformance can be obtained by adequate demonstration of supplier's capabilities in processes (design, development, …)

- ISO registration by an officially accredited body, re-registration every three years

Capability Maturity Model (CMM)

- **Initial level: software development is ad-hoc**
- **Repeatable level: basic processes are in place**
- **Defined level: there are *standard* processes**
- **Quantitatively managed level: data is gatheread and analyzed routinely**
- **Optimizing level: stable base, data is gathered to improve the process**

CMM: critical notes

- Most appropriate for big companies
- Pure CMM approach may stifle creativity
- Crude 5-point scale (now: CMMI)

Get started on Software Process Improvement (SPI)

- Formulate hypotheses
- Carefully select metrics
- Collect data
- Interpret data
- Initiate improvement actions

- Iterate
-

Lessons w.r.t. data collection

- Closed loop principle: result of data analysis must be useful to supplier of data
- Do not use data collected for other purposes
- Focus on continuous improvement
- Only collect data you really need

Summary

- Product quality versus process quality
- Quality conformance versus quality improvement
- Quality has to be actively pursued
- There are different notions of quality
- Quality has many aspects
- Quality is hard to measure