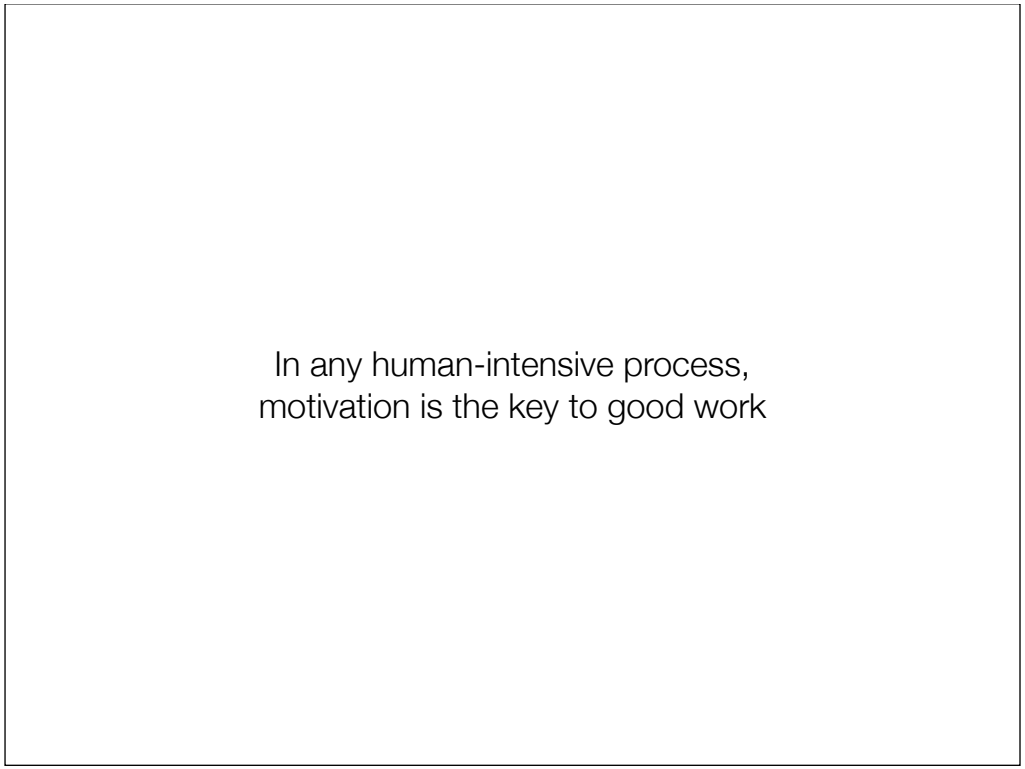
Managing Software Process Quality

Alexandre Bergel
abergel@dcc.uchile.cl

Product quality is a key measure of the software process



To consistently achieve superior performance, management must establish challenging quality goals and strive to meet them. Conversely, if senior management tolerates poor work, sloppiness will pervade the entire organization. This is true not just for complex software products, but for all aspects of a business.

When meeting are perpetually late, status reports are inaccurate, or management's memos have typographical errors, the programmers realize that quality is not a priority.

Basi

Basic quality principle

Basic quality principles

- Unless you establish aggressive quality goals, nothing will change
- If these goals are not numerical, the quality program will remain just talk
- Without quality plans, only you are committed to quality
- Quality plans are just paper unless you track and review them

The quality management paradigm

The basic principles of software quality management are much like those for cost management

1 - you set goals

2 - make plans

3 - track performance

4 - adjust the plan

Experience by Poston and Bruen

Studied a RT monitoring program of 27,719 LOC

By applying a quality plan

- failure density rates reduced from 1.3 to 0.072 defects/1000 LOC
- 95% improvement
- productivity rate up to 29 LOC/day/man from an average of 7

R. M. Poston, M. W. Bruen, “Counting Down to Zero Software Failures” IEEE Software archive Volume 4 , Issue 5, Pages 54-61, Year of Publication: 1987, ISSN:0740-7459

Measuring quality

Measuring quality

People can only respond to a few motivational drives at a time

Need to establish a small number of specific, numerical quality measures

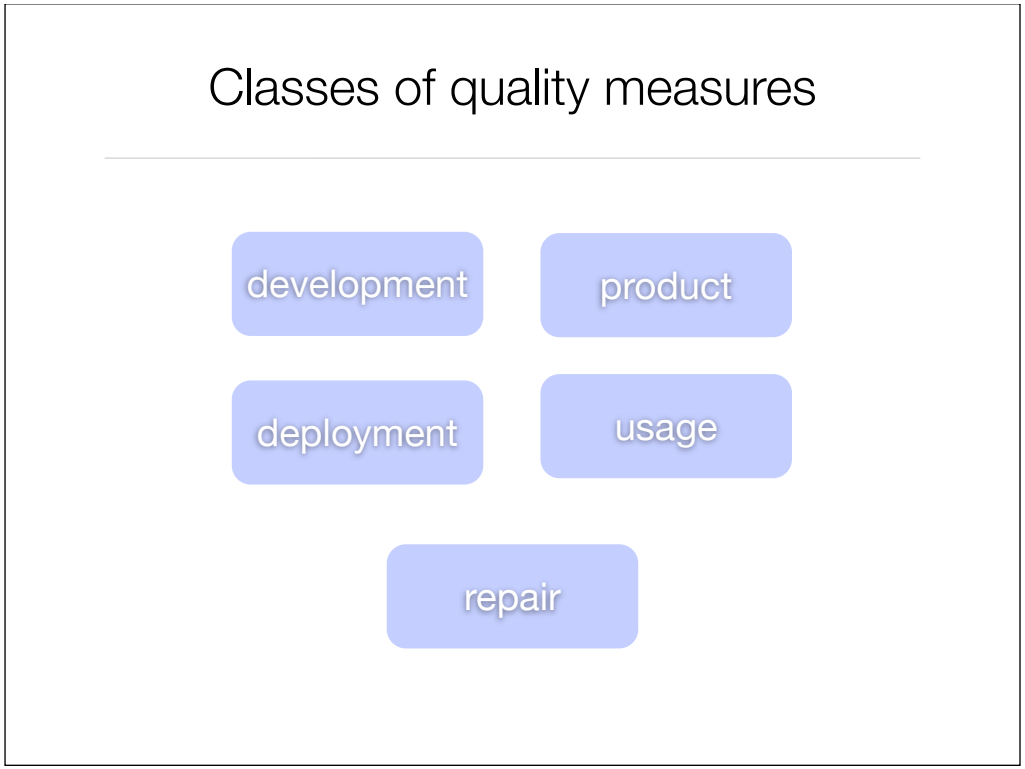
Importance of numerical measure

Without numerical measure, schedule remains the only measure for development

When adhering to schedule is the only sign of progress, all energies are directed toward meeting deadline

Numerical quality measure

- No single measure can characterize a complex product
- Too many measures can be confusing
- Need to use a few carefully selected measures as quality indicators



Quality measures fall into the following general classes:

- development : defects founds, change activity
- product: defects found, software structure, information (documentation) structure, controlled tests
- acceptance: problems, effort to install, effort to use
- usage: problems, availability, effort to install, effort to use, user opinions
- repair: defects, resources expended

Measurement criteria

Deciding what measures to use is essential

Defect measures

- Defect counts not strongly connected to customer satisfaction
- Determining what truly represents quality for the customer is not simple

Change activity

May be a useful measure of development quality

When high in a development program, it indicates overall quality problems

Change activity can be a useful measure of development quality. When change activity remains high late in a development program, it is a good indication of overall quality problems.

Controlled tests

Tests provide a simulated work environment

Regression testing cannot be avoided these days