The Use of Software Metrics for Enhancing Software Quality and Reliability

November 15, 2000
Richard Huntrods

Presentation Overview

- Introduction
- A Brief History
- Contemporary State of the Art
- Future Perspectives
- Conclusion

Introduction (1)

- Goals:
  - We want software that meets user requirements
  - Software quality and reliability are critical
- Software metrics are a fundamental tool
- Metrics are measurement and prediction

Introduction (2)

- Object-oriented metrics necessary for OO code
- Industry data extends current student data models
- Statistical Analysis of the data is the next step
- Ongoing tool development to automate processes

A Brief History (1)

- Metrics framework:
  - classifying entities to be examined
  - determining relevant measurement goals
  - identifying process maturity

A Brief History (2)

- Classifying Entities:
  - Process
  - Products
  - Resources
A Brief History (3)

- Process
  - duration, effort, incidents
- Products
  - external vs. internal measures
- Resources
  - persons, materials, methods

A Brief History (4)

- Relevant measurement goals
  - Goal Question Metric (GQM)
- Process Maturity
  - Capability Maturity Model (CMM)

A Brief History (5)

- Software measurement validation
  - Measurement systems
    - LOC/hr., defects/KLOC
  - Prediction systems
    - how does quality relate to the measures?
    - Is it possible to predict quality from measures?

Contemporary State of the Art (1)

- Standard Metrics
  - LOC
  - McCabe’s cyclomatic complexity
  - fan-in, fan-out
  - Halstead volume

Contemporary State of the Art (2)

- OO Concepts
  - Objects (classes)
  - Encapsulation
  - Inheritance
  - Polymorphism

Contemporary State of the Art (3)

- OO Metrics - Chidamber and Kemerer
  - Weighted methods per class
  - depth of inheritance tree
  - number of children
  - coupling between objects
  - response for a class
  - lack of cohesion in methods
Contemporary State of the Art (4)

• Current models employ student data
  – limited scope
  – limited developer experience
  – limited program complexity
• New Models based on Industrial data
  – “real world” scope
  – “real world” developer experience
  – “real world” program complexity

Future Perspectives

• Predictive models
  – use statistical methods to validate current measures
  – produce predictive models and history match with existing measures

Conclusion (1)

• History
  – Software quality and reliability are the goal
  – Software Metrics help achieve the goal
• Contemporary State of the Art:
  – Conventional and OO metrics are required
  – Metrics provide measures and prediction
  – Industrial data allows more meaningful/useful models

Conclusion (2)

• Future Perspectives:
  – Statistical methods for validation of current models
  – Statistical methods for production of predictive models
• Questions