

Object-Oriented Software Development

CS151

Chris Pollett

Aug. 24, 2005.

Outline

- Challenges of Software Development
- Software Development Activities
- Software Development Process
- Desirable Qualities of Software Systems
- Where Software Development and Engineering Differ
- Object Orientation
- Some Common Software Processes

Challenges of Software Development

- Complexity -- programs now are getting very large and complex - even millions of lines.
- Longevity and Evolutions -- software products can be in use for decades.
(SABRE)
- High User Expectations -- mistakes can lead lost business, lawsuits, etc.

Software Development Activities

- Requirements Analysis
- Design
- Implementation and Unit Testing
- Integration and System Testing
- Maintenance

Desirable Qualities of Software Systems

- For the end user:
 - Usefulness, Timeliness, Reliability, Maintainability, Reusability, User-friendliness, Efficiency
- It is hard to achieve all of these simultaneously with low cost.

Reasons to try to improve maintainability:

- Software can be around a long time so total maintenance costs can be higher than development costs
- Reliability usually happens only after several releases of software. To be able to make new releases requires maintainable code.
- High maintainability requires flexibility in design, which can enhance the reliability, usefulness, and the ability to contain costs.

Factors Contributing to Maintainability

- Flexibility -- how easy it is to change the software system
- Simplicity -- how easy it is for people to avoid mistakes when using the software system
- Readability -- how easy it to understand the code and its accompanying documentation.

Where Software Development and Engineering Differ

- Analysis of Designs -- historically, designs have been improved by craftsmanship: by looking at what works and what doesn't in old designs. Engineering can make use of mathematical models to calculate how good a design is without relying on as much trial and error.
- Nonrecurrence of failures. In well established engineering fields mistakes are rarely repeated. In software engineering its very hard to ensure mistakes are not repeated.
- Codification of Knowledge -- in engineering there is a large accumulation and reuse of prior solutions compare to in software engineering.

Object Orientation

- A software system consists of two kinds of components: models that store data which are a representations of a pertinent parts of the real world, and an algorithms which capture the computations involved in manipulating or processing the model.
- Prior paradigms for software systems such as *control-flow* of the 50s and 60s and *data flow* of the 70s and 80s emphasized one either data or algorithm at the expense of the other.
- The object oriented approach takes a balanced approach to software systems

Some Common Software Processes

- Spiral model
- Iterative process. This is split into phases following the software activities we listed before: analysis and modeling, design, implementation, maintenance. Once one iteration is done we repeat this process.
- The iterative process has also inspired refinements:
 - Rational Unified Process (RUP)
 - Extreme Programming