COMPSCI 320 Introduction to Software Engineering:

Third-Party Software

Slides adapted from Gordon Anderson

One-Slide Summary

- Software teams decide:
  - What to write themselves
  - What to use **third-party software** for
- How do we **assess the tradeoffs** of different options?

Third-Party Software

- “Don’t reinvent the wheel”
- Make sure you choose the right wheel…

Benefits of Software Reuse

1. Saves development time and effort
2. Rely on others' domain expertise
3. Already tested* (be careful here!)
4. Support and documentation.

Considerations

1. What exists already?
2. Cost of building from scratch?
3. Can it be proprietary?
4. Support/compatibility guarantees?
5. Team familiarity/learning curve?
6. Documentation?
7. Simplicity vs. Flexibility
8. Signs of active development

Selection Strategy

1. Search for several candidate libraries
2. Implement demos with candidates
   - Fixed set of features
3. “Bake off”:
   - Evaluate with fixed criteria
   - Set a due date for the decision.

Selection Strategy

Scenario for discussion: A non-profit organization is updating their website.

Committee designing & implementing new website has mostly non-technical members and one or two technical members.

What technologies should we use?

Summary

- Software reuse has **benefits** & **pitfalls**
- **Consider tradeoffs** when evaluating third-party options
- Outlined a **strategy for examining** potential options

(If time) Info Dump About Licensing

Software Licensing and Related Terms

Software license:

A legal document that specifies the terms under which the software may be used and redistributed. A license may be modeled on an existing license or be very specifically written by the creator of the software.

Software Licensing and Related Terms

- ***Open-source*** is a development model. Source code may be used, modified and distributed.
- Some open-source may not be used for profit.
- Examples of open software licenses often used as a model: Berkeley Software Distribution ("BSD"), Apache, MIT-style (Massachusetts Institute of Technology), or GNU General Public License ("GPL").
- ***Closed source:*** use restricted, proprietary: must pay for the use of the software.

Software Licensing and Related Terms

Intellectual Property (IP):

"intangible creations of the human intellect". The legal rights to this type of property are controlled mostly by copyrights, patents, and trademarks.

Scenario: if you are working for a company and create software- who owns the IP? Generally the company, although in many cases there is joint ownership. If you develop software of your own design but use company equipment, the company may own the IP.

Software Licensing and Related Terms

Copyright:

A legal right of the creator of an original work to control how their work is used. Covers the "expression" of the original idea, not the idea itself. A ***patent*** would cover the idea itself.

All software is copyright protected in the USA- other countries have their own laws.

Software Licensing and Related Terms

Copyright vs. Patent:

Example: a piece of software is the expression of an idea- an algorithm. How the software can be used is governed by ***copyright*** law. A user could try and discover the algorithm behind the software and write their own implementation of it without infringing on the copywrite. However, if the algorithm was ***patented***, then the re-implementation would violate the patent.