The challenges of creating open source education software: the Gild experience

Daniel M. German
Work with Peter Rigby, Davor Cubranic, Margaret-Ann Storey, Suzanne Thomson

University of Victoria
Canada
What is Gild

- Groupware-enabled Integrated Learning and Development (GILD) Environment
- Eclipse perspective designed to improve learning and teaching of programming
- Collaborative effort between Universities
- Used by over 300 first-year students at UVic
  - Used in Canada, USA, and Germany
package pirates;

public class MainPirates{

    public static void main(String [] args){
        System.out.println("Pirates!");
        Captain hook = new Captain();
        hook.speak();
        Deckhand sмеe = new Deckhand();

        Deckhand blackbeard = new Deckhand();

        // Does the output above correctly indicate how many
        // swords have been given out?
        // Why/Why not?

        // Unlike the Shape class from lecture, this
        // statement is not valid.
        // Why? (Hint: Look at the Pirate class.)

        // Pirate p = new Pirate();

    }
}

This week, we also have a couple of code examples that you can run and play with in Gild. See the clock and pirates directories in the activities directory. There are a few questions scattered throughout to point out key concepts. Take as much or as little time as you like, if you get stuck, refer to your notes and textbook, and then try again!

1. Open up, explore, and answer the questions we posed within the code for the Pirates, Captain, Deckhand and MainPirates classes in Gild. Course concepts covered: Usage modifiers, super, inheritance
   - Pirates.java
   - Captain.java
Eclipse features

- Three main users
  - Students
    - A simplified user interface targeting novices
  - Instructors
    - Support to deliver lectures using eclipse
  - Markers
    - Support to automatize marking tasks
Benefits of being open source

- Eclipse is open source
  - People expect plugins/perspectives to be open source
  - We are not interested in selling it
  - We are interested in the research issues

- Creates interest and attracts users
  - Several universities are showing interest
  - Mentioned in the “Eclipse for Dummies” book
  - We are still waiting for patches!
Challenges in creating a development community

- Users are not usually skilled Java developers
  - Novice programmers, instructors, TAs
- Developers
  - Paid vs. not paid
- Researchers
  - Professors and graduate students
Research, Development, and Maintenance

- Maintenance is not research
  - Need a community to help with maintenance
  - Our research group will develop it as long as it helps our research

- But developers are not users of the core Gild product (the student perspective)
  - Other ways of attracting contributors

- How can we attract a community of contributors?
Community Building (Novices)

- Novices can’t even use Eclipse
- Once they learn it, they have few incentives to contribute to Gild
  - They don’t need it, hence no scratch your own itch
- One potential reason: they want to learn how to program plugins in Eclipse.
Community Building (Instructors)

- Instructor can pick tool for class
  - It is always a high risk commitment
  - Offer incentives:
    - Bundle Course content with Gild
    - First year Programming course in Java
  - Instructors produce good bug reports
  - But tool can’t break in the middle of the term
    - Critical mission!
Community Building (Teaching Assistants)

- Marking Prospective
  - Quicker, more consistent marking
  - TAs scratching own itch
- But, can we motivate TAs to contribute to other areas of Gild?
Business Community

- Interest from IBM and Microsoft
  - How do novices use an IDE (Requirements)
  - Papers and conferences
  - Funding from IBM (Eclipse Innovation grant)
    - Major source of Gild funding
Finding a maintainer

- A good maintainer is key to the success of a F/OSS project
- As Gild evolves, it becomes more difficult to maintain it
Finding a maintainer...

- Faculty
  - Too busy with other endeavours
  - Not necessarily the most skilled
- Employees
  - Need to get paid!
- Master’s students
  - Too volatile: they are gone too quickly
Finding a maintainer…

- PhD students are better suited
  - Can we find ways to mix the job of being a maintainer with their research project?
Conclusion

- Open source projects and Research projects have different goals
  - In OSS *code* is paramount not the *idea*
  - OSS is usually built and maintained by its users, not by those who believe in it

- In order to create a community you need *skilled* people that need your product
Thank You!

- Questions?
- Comments?
- Gild Website: http://gild.cs.uvic.ca/
- Email: dmg@cs.uvic.ca
package pirates;

public class MainPirates{

    public static void main(String [] args){
        System.out.println("Pirates!");

        Captain hook = new Captain();
        hook.speak();
        Deckhand smee = new Deckhand();

        Deckhand blackbeard = new Deckhand();

        // Does the output above correctly indicate how swords have been given out?
        // Why/Why not?

        // Unlike the Shape class from lecture, this statement is not valid.
        // Why? (Hint: Look at the Pirate class.)

        // Pirate p = new Pirate();
static void main(String [] args){
    System.out.println("Pirates!");
    Captain hook = new Captain();
    hook.speak();
    Deckhand smee = new Deckhand();
    Deckhand blackbeard = new Deckhand();

    Does the output above correctly indicate how many swords have been given out?
    Why/Why not?

    Unlike the Shape class from lecture, this statement is not valid.
    Why? (Hint: Look at the Pirate class.)

    Pirate p = new Pirate();
This week we also have a couple of code examples that you can run and play with in Gild. See the clock and pirates directories in the activities directory. There are a few questions scattered throughout to point out key concepts. Take as much or as little time as you like, if you get stuck, refer to your notes and textbook, and then try again!

1. Open up, explore and answer the questions we posed within the code for the Pirates, Captain, Deckhand and MainPirates classes in Gild. Course concepts covered: Usage modifiers, super, inheritance

- **Pirates.java**
- **Captain.java**
- **Deckhand.java**
- **MainPirates.java**

Extra Help

You are trying to make an instance of an abstract class or of an interface. Abstract classes and interfaces cannot be instantiated. To use an abstract class you must create a subclass of the abstract superclass.

- e.g. In file Shape.java:
  ```java
  public abstract class Shape {...}
  ```
- In file Square.java:
  ```java
  public class Square extends Shape {...}
  ```