

DRY Principle

Gabor Bata - June 2, 2014



DRY - Don't Repeat Yourself

“Every piece of knowledge must have a single, unambiguous, authoritative representation within a system.”

Andy Hunt and Dave Thomas
The Pragmatic Programmer

Why is it important?

Knowledge changes rapidly, regular maintenance is needed

- requirement changes
- new rules (e.g. government changes)
- bug in the system have to be corrected
- etc.

The Evils of Duplication (four *i*'s)

- **Imposed duplication**
it seems there are no other choice (project standards)
- **Inadvertent duplication**
developers don't realize if they causing duplication
- **Impatient duplication**
developers get lazy
- **Interdeveloper duplication**
multiple people/team duplicates information

Imposed duplication

- Don't duplicate information in documentation and code
 - bad code requires lots of comments
 - outdated comments are worse than no comments
- Automate repetitive tasks
 - code generation (from schema or documents), continous build, automated tests, BDD etc.

Inadvertent duplication

Duplication comes from bad design, improper performance improvements etc.

Example:

```
public class Line {  
    public Point start;  
    public Point end;  
    public double length;  
}
```

Impatient duplication

- Causing by time pressure, laziness of developers, *copy-and-paste*
- Shortcuts make for long delays
- Use abstraction instead of *if-then* and *switch-case logic*
 - needs time in short term but saves pain later
 - design patterns

Interdeveloper duplication

- Different developers on a project could implement the same thing many times
- To deal with that you need
 - frequent communication among developers/teams
 - read other people's source code and documentation
 - make things easy to reuse

Questions



Thanks for your attention!

Now let's have some
Canada Dry!

