# Don't repeat yourself

From Wikipedia, the free encyclopedia
(Redirected from DRY code)

*Contrast with redundancy and mirror.*

In software engineering, **Don't Repeat Yourself** (**DRY**) or **Duplication is Evil** (**DIE**) is a principle of software development aimed at reducing repetition of information of all kinds, especially useful in multi-tier architectures. The DRY principle is stated as "Every piece of knowledge must have a single, unambiguous, authoritative representation within a system." The principle has been formulated by Andy Hunt and Dave Thomas in their book *The Pragmatic Programmer*. They apply it quite broadly to include "database schemas, test plans, the build system, even documentation."[1] When the DRY principle is applied successfully, a modification of any single element of a system does not change other logically-unrelated elements. Additionally, elements that are logically related all change predictably and uniformly, and are thus kept in sync. Besides using methods and subroutines in their code, Thomas and Hunt rely on code generators, automatic build systems, and scripting languages to observe the DRY principle across layers.

## Contents


- 1 Applying DRY
- 2 See also
- 3 References
- 4 External links


## Applying DRY

Also known as Single Source of Truth, this philosophy is prevalent in model-driven architectures, in which software artifacts are derived from a central object model expressed in a form such as UML. DRY code is created by data transformation and code generators, which allows the software developer to avoid copy and paste operations. DRY code usually makes large software systems easier to maintain, as long as the data transformations are easy to create and maintain. Tools such as XDoclet and XSLT are examples of DRY coding techniques. Examples of systems that require duplicate information are Enterprise Java Beans version 2, which requires duplication not just in Java code but also in configuration files. Examples of systems that attempt to reduce duplicate information include the Django web framework, Ruby on Rails application development environment and Enterprise Java Beans version 3.

Kent Beck has mentioned that he does not "subscribe to [DRY] for test code because [he] want[s] [his] tests to read like a story".[2]

## See also

- Abstraction principle (programming)
- Code reuse
- Single Source of Truth
- Occam's Razor
- Separation of concerns

- KISS principle
- You Ain't Gonna Need It
- Rule of three (programming)

# References

1. ^ Dave Thomas, interviewed by Bill Venners (2003-10-10). "Orthogonality and the DRY Principle" (http://www.artima.com/intv/dry.html) . http://www.artima.com/intv/dry.html. Retrieved 2006-12-01.
2. ^ Kent Beck, interviewed by Martin Lippert (2010-09). "Software Engineering Radio Episode 167: The History of JUnit and the Future of Testing with Kent Beck" (http://www.se-radio.net/2010/09/episode-167-the-history-of-junit-and-the-future-of-testing-with-kent-beck) . http://www.se-radio.net/2010/09/episode-167-the-history-of-junit-and-the-future-of-testing-with-kent-beck. Retrieved 2011-01-25.

# External links

- Orthogonality and the DRY Principle (http://www.artima.com/intv/dry.html)
- c2.com the original (http://c2.com/cgi/wiki?DontRepeatYourself)
- Once and Only Once (c2.com) (http://c2.com/cgi/wiki?OnceAndOnlyOnce)
- 97 Things Every Programmer Should Know (O'Reilly) (http://programmer.97things.oreilly.com/wiki/index.php/Don't_Repeat_Yourself)
- Codereflect.com - DRY Principle (http://codereflect.com/2010/10/03/dry-dont-repeat-yourself/)

Retrieved from "http://en.wikipedia.org/wiki/Don%27t_repeat_yourself"
Categories: Software development philosophies | Programming principles

- This page was last modified on 25 January 2011 at 21:49.
- Text is available under the Creative Commons Attribution-ShareAlike License; additional terms may apply. See Terms of Use for details.
  Wikipedia® is a registered trademark of the Wikimedia Foundation, Inc., a non-profit organization.