

Sumário

1	Introdução	1
1.1	Organização do Trabalho	1
2	DRY	2
2.1	Princípios relacionados	2
2.1.1	Princípio Open/Closed	2

Lista de Figuras

Lista de Tabelas

Capítulo 1

Introdução

Uma propriedade intrínseca dos softwares em um ambiente real é a sua necessidade de evolução[?]. Apesar dessa necessidade ser uma característica própria, a manutenção e a evolução de software são caracterizadas pelo seu grande custo e lenta velocidade de implementação[?]. Isso faz necessário a busca de ferramentas e técnicas que permitam a redução da complexidade através do aumento incremental da qualidade interna do software[?].

Muitos princípios e práticas foram criados com o propósito de controlar a duplicação. Considere, por exemplo, que todas as formas normais de banco de dados de Codd servem para eliminar a duplicação de dados. Considere também como a orientação orientada a objetos serve para concentrar o código classes bases que de forma contrária seriam redundantes. Programação estruturada, programação orientada a aspectos, programação orientada a componentes, são todas, em parte, estratégias para eliminar a duplicação[Martin, 2009].

Nesse trabalho iremos examinar o princípio Don't Repeat Yourself. Em [Henney, 2010] Steve Smith afirma que este é talvez um dos mais fundamentais princípios da programação. Ele foi formulado por Andy Hunt e Dave Thomas em “The Pragmatic Programmer”, e permeia muitos outros padrões e práticas do desenvolvimento de software[Henney, 2010].

Durante este trabalho iremos apresentar o princípio e alguns dos conceitos relacionados, discutir a duplicação e por fim apresentar as conclusões geradas pelo trabalho.

1.1 Organização do Trabalho

Capítulo 2

DRY

Em [Martin, 2009], Robert C. Martin defende a importancia da programação dizendo que apesar das ferramentas de geração de código, nós nunca estaremos livres do código. Ainda segundo ele, o código representa o detalhamento dos requisitos, sendo a linguagem utilizada para expressá-los.

Infelizmente, conhecimento não é estável, ele muda constantemente. Essa instabilidade faz com que uma grande parte do tempo de desenvolvimento seja gasto na manutenção do código, reorganizando e re-expressando o conhecimento do sistema.

Ao realizar manutenções é necessário encontrar e mudar a representação do conhecimento incorporado a aplicação. O problema é que conhecimento é fácil de ser duplicado em especificações, processos e programas que nós desenvolvemos[Hunt et al., 1999].

Essa duplicação incha desnecessariamente a base de código, resultando em mais oportunidades para bugs e adicionando complexidade adicional para o sistema. O inchaço que a duplicação traz ao sistema também o torna mais difícil de ser totalmente compreendido pelos desenvolvedores do sistema, ou de ter certeza que certas mudanças feitas em um lugar não precisam ser feitas em outras partes que duplicam a lógica na qual se está trabalhando[Henney, 2010].

Nesse contexto é que encontramos o princípio DRY que diz que todo pedaço de conhecimento deve ter uma única, universal e autoritativa representação dentro de um sistema.

Quando seguido respeitando a estrutura, lógica, processo e função, o princípio DRY fornece uma orientação fundamental para os desenvolvedores de software e auxilia a criação de aplicações mais simples, manuteníveis e de alta qualidade.. [Henney, 2010].

2.1 Princípios relacionados

Existem uma série de outros princípios de software relacionados ao princípio DRY. A seguir iremos apresentar alguns deles e falar sobre como esses princípios se relacionam.

2.1.1 Princípio Open/Closed

Esse princípio foi cunhado em 1988 por Bertran Meyer[?]

Referências Bibliográficas

- [Henney, 2010] Henney, K. (2010). *97 Things Every Programmer Should Know: Collective Wisdom from the Experts*. O'Reilly Media, Inc., 1st edition.
- [Hunt et al., 1999] Hunt, A., Thomas, D., and Cunningham, W. (1999). *The Pragmatic Programmer. From Journeyman to Master*. Addison-Wesley Longman, Amsterdam.
- [Martin, 2009] Martin, R. C. (2009). *Clean Code: A handbook of agile software craftsmanship*. Prentice Hall.