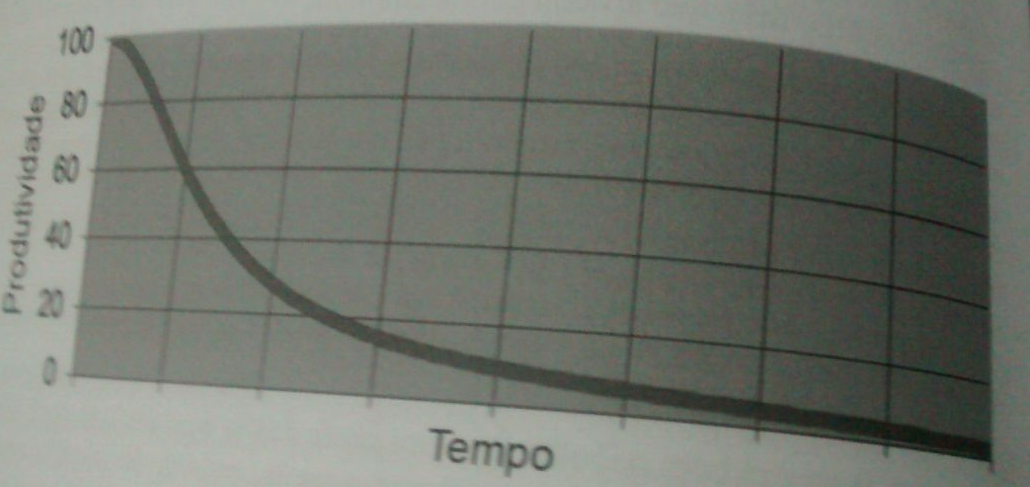
**CÓDIGO LIMPO**

**Autor Livro**: *Robert Cecil Martin, também conhecido como "Uncle Bob" (Tio Bob em português), é uma grande personalidade da comunidade de desenvolvimento de software, métodos ágeis e software craftsmanship, atuando na área desde 1970. Atualmente é consultor internacional e autor de vários livros abordando o tema. Uncle Bob foi um dos 17 signatários originais do Agile Manifesto em 2001. É fundador da Object Mentor Inc, uma empresa de consultoria especializada em C++, Java, OOP, design patterns, UML, metodologias ágeis, e Extreme Programming. Atualmente faz parte da equipe 8th Light.*

**Autor resumo**: Jackson Smith.

**Capitulo 1:**

**O Código ruim:** Para Robert um bom código importa, ele cita a história de um aplicativo criado na década 1980 de sucesso, mas teve que se encerrou devido a manutenção do código não ser possível devido à dificuldade de entender e alterar o código. Um código ruim gera atraso, e dos motivos é a pressa, mas esse “ganho” a longo prazo poderá ter um custo de tempo alto na manutenção e implementação de novas funcionalidades, importante também não deixar para mais tarde pois *Mais tarde é igual a nunca*.

**O custo de ter um código confuso**: O gráfico abaixo mostra que conforme a confusão vai aumentando a produtividade vai diminuindo, até que fica impossível continuar o desenvolvimento.  
**O grande Replanejamento**: Quando um projeto é de difícil leitura ou continuidade apenas replanejamento não é uma boa medida, Robert já presenciou que times já levaram 10 anos pare relampejarem um projeto, e ao final deste tempo já era necessário um outro replanejamento.

**Atitude:** Desenvolvedores não devem deixar que a pressa resulte em um código ruim, Robert faz uma analogia a um cirurgião e um paciente, se um paciente exige que o médico não fizesse a higienização das mãos antes da operação para que o procedimento cirúrgico fosse mais ágil, é de se esperar que complicações devido a infecção possam ocorrer. Os desenvolvedores são os principais responsáveis pela qualidade de um bom código

**Código Limpo:** Dentre os diversos autores que deram sua contribuição para definir esta habilidade, as citações convergem para um código que seja interpretado da forma mais natural possível semelhante a uma conversa entre duas pessoas, que contenham testes unitários, que tenha bons nomes para seus métodos, variáveis, classes eetc. que não possuam repetições e que as abstrações utilizadas em metodologia de orientação a objetos sejam utilizadas.