

Aluno: Bernardo de Resende Marcelino

Disciplina: Projeto de software

Professor: João Paulo Aramuni

Resenha Crítica "On the Criteria To Be Used in Decomposing Systems into Modules"

O artigo "On the Criteria To Be Used in Decomposing Systems into Modules" de D.L. Parnas busca estabelecer critérios fundamentais para a decomposição eficaz de sistemas de software em módulos, questionando as abordagens convencionais de modularização e propondo uma metodologia alternativa baseada no conceito de "information hiding", demonstrando através de exemplos práticos como diferentes critérios de decomposição podem impactar significativamente a flexibilidade, compreensibilidade e manutenibilidade de sistemas de software.

Parnas identifica que, embora a programação modular seja amplamente reconhecida como benéfica para o desenvolvimento de software, proporcionando vantagens gerenciais através da possibilidade de desenvolvimento paralelo por equipes independentes, flexibilidade de produto permitindo mudanças drásticas em módulos individuais sem afetar outros componentes, e compreensibilidade facilitando o estudo do sistema uma parte por vez, raramente são discutidos os critérios específicos que devem orientar a divisão de um sistema em módulos, resultando em decomposições que frequentemente seguem apenas a intuição ou convenções estabelecidas sem fundamentação teórica sólida.

Através do exemplo detalhado de um sistema KWIC (Key Word in Context), o autor apresenta duas abordagens contrastantes de modularização, onde a primeira, representativa da metodologia convencional, organiza módulos baseados nos passos sequenciais de processamento - entrada, deslocamento circular, alfabetização, saída e controle mestre - seguindo essencialmente a lógica de um fluxograma e criando interfaces complexas entre módulos que compartilham estruturas de dados e formatos específicos, enquanto a segunda decomposição, fundamentada no princípio de "information hiding", organiza módulos em torno de decisões de design que devem ser ocultadas dos demais componentes, como armazenamento de linhas, entrada, deslocamento circular, alfabetização, saída e controle, onde cada módulo esconde aspectos específicos da implementação através de interfaces abstratas e bem definidas.

A análise comparativa das duas abordagens revela que, embora ambas possam produzir sistemas funcionalmente equivalentes, a decomposição baseada em "information hiding" demonstra superioridade significativa em termos de flexibilidade para mudanças, uma vez que alterações em decisões de design como formato de armazenamento, empacotamento de caracteres ou estratégias de indexação ficam confinadas a módulos específicos na segunda abordagem, enquanto na primeira tais mudanças se propagam através de múltiplos módulos devido ao compartilhamento de estruturas de dados, além de facilitar desenvolvimento independente através de interfaces mais abstratas que permitem trabalho paralelo mais cedo no processo de desenvolvimento, e melhorar compreensibilidade ao isolar responsabilidades específicas em módulos autocontidos.

Parnas estabelece como critério fundamental que cada módulo deve ser caracterizado pelo conhecimento de uma decisão de design que esconde de todos os outros, propondo exemplos específicos como estruturas de dados com seus procedimentos de acesso e

modificação devem formar módulos únicos, sequências de instruções para chamadas de rotinas devem ser agrupadas com as próprias rotinas, formatos de blocos de controle em sistemas operacionais devem ser ocultados em módulos dedicados, códigos de caracteres e ordenações alfabéticas devem ser isolados para máxima flexibilidade, e sequências de processamento devem ser escondidas sempre que possível para facilitar mudanças futuras, reconhecendo que embora essa abordagem possa resultar em implementações menos eficientes devido ao overhead de chamadas entre módulos, técnicas adequadas de implementação podem mitigar esses problemas através de inserção inteligente de código e ferramentas especializadas de montagem.

Dessa forma, o artigo de Parnas apresenta uma contribuição fundamental para a engenharia de software ao formalizar o conceito de "information hiding" como critério central para decomposição modular, demonstrando que a organização de sistemas baseada em fluxogramas e passos de processamento, embora intuitiva, resulta em arquiteturas frágeis e difíceis de manter, enquanto a abordagem orientada ao ocultamento de decisões de design produz sistemas mais robustos, flexíveis e compreensíveis, estabelecendo princípios que continuam relevantes para o desenvolvimento de software moderno e influenciando conceitos posteriores como encapsulamento e abstração em programação orientada a objetos.