

O artigo “On the Criteria to Be Used in Decomposing Systems into Modules”, escrito por David L. Parnas em 1972, é considerado um marco na história da engenharia de software. Nele, o autor discute a questão de como dividir um programa grande em módulos menores, apresentando uma proposta inovadora que transformou a forma de pensar o desenvolvimento de sistemas. A questão central levantada por Parnas é a seguinte: qual é a maneira mais eficiente de organizar um sistema em partes que sejam independentes, mas que ainda trabalhem de forma integrada?

Na época, a forma mais comum de modularizar programas era basear-se em fluxogramas, isto é, dividir o software de acordo com a sequência de etapas que ele executava. Por exemplo, criava-se um módulo para a leitura dos dados, outro para o processamento e um terceiro para a impressão dos resultados. Essa organização, que o autor chama de “convencional”, parecia intuitiva, mas apresentava sérios problemas. Como todos os módulos precisavam conhecer detalhes internos da implementação, qualquer mudança em um deles exigia alterações em vários outros, tornando o sistema difícil de manter e de evoluir.

O autor propõe, o princípio do ocultamento de informação (*information hiding*). Nesse modelo, cada módulo deve ser responsável por esconder uma decisão de projeto importante do restante do sistema. Em vez de organizar os módulos de acordo com a sequência de operações, eles são estruturados de modo que cada um guarde “segredos” específicos, como a forma de armazenar dados ou a escolha de determinado algoritmo. Assim, mudanças internas em um módulo não afetam os demais, pois estes apenas interagem com ele por meio de interfaces bem definidas.

Para ilustrar sua proposta, o autor utiliza o exemplo de um programa de geração de índices de textos (KWIC). Na abordagem convencional, todos os módulos dependem da forma como os dados são armazenados, o que gera fragilidade. Já na proposta baseada no ocultamento de informação, apenas um módulo conhece os detalhes do armazenamento. Os demais apenas requisitam informações a esse módulo, sem precisar saber como elas estão estruturadas. Isso garante maior independência e reduz a propagação de mudanças ao longo do sistema.

A principal vantagem dessa visão está na facilidade de manutenção. Alterações em um aspecto específico, como a forma de armazenamento de dados, exigem modificações apenas no módulo correspondente, sem impacto nos outros. Além disso, esse modelo favorece o trabalho em equipe, pois diferentes desenvolvedores podem atuar em módulos distintos sem a necessidade de conhecer os detalhes internos uns dos outros. Também se torna mais simples compreender partes isoladas do sistema, sem precisar abarcar sua totalidade de uma só vez.

Em conclusão, o artigo de Parnas apresenta uma ideia simples, mas profundamente transformadora. Em vez de pensar na decomposição de sistemas como um conjunto de etapas operacionais, o autor propõe que se identifiquem as decisões de projeto mais delicadas ou suscetíveis a mudanças, criando módulos específicos para encapsular essas decisões. Esse princípio do ocultamento de informação tornou-se um dos fundamentos da engenharia de software moderna e influenciou diretamente práticas como a programação orientada a objetos e o desenvolvimento de sistemas modulares.