

Gustavo Santos Teixeira
Matrícula[202200149]

Resumo do artigo:

Com base nas informações fornecidas no artigo, há uma boa retratação do modo como a Engenharia de Software se desenvolveu, principalmente pelo seu conceito de “desenvolvimento” nos anos iniciais, que era profundamente integrado à ideia de avanço no hardware, não tendo um foco específico no desenvolvimento do software. Outro ponto que desperta interesse, foi o fato dele dar ênfase em observar tanto os erros quanto os acertos do passado, para utilizá-los no desenvolvimento, pois este é um ponto que muitas vezes não é levado em consideração e muda totalmente o resultado de um processo de pesquisa.

Posteriormente, a necessidade de novos softwares foi aumentando, pessoas de diversas áreas começaram a migrar para essa área, que antes estava contando com uma abundância de matemáticos e pessoas relacionadas às ciências computacionais, aumentando a diversidade de modos de pensar com relação ao processo de criação de sistemas satisfatórios às demandas existentes. Com isso, a característica de programar e corrigir foi se instalando, tendo certa influência do processo de design após esta etapa, buscando cada vez mais uma melhor organização do código, que levaria a uma melhor performance.

Com isso houve o desenvolvimento da programação estruturada, tendo dois ramos principais, sendo um mais focado na formalidade e o outro em uma mistura menos formal de meios técnicos. Através do sucesso da programação estruturada, surgiram novas abordagens desse modo de construir algoritmos, um bom exemplo foi a aplicação da modularização, que é dividir o programa em partes menores que realizam tarefas específicas. Podemos resumir essa fase como um período de programar, corrigir, buscar melhores formas de estruturar o código, aumentando ainda mais a sua eficácia.

A preocupação com as correções de erros, modos de fazer melhorias e evitar falhas continuou aumentando gradativamente com o passar do tempo. Devido ao grande número de processos envolvidos na criação de um software, a preocupação com os requisitos e design foi deixada de lado, levando a tentativas de melhorar a produtividade de software e o uso de melhores práticas para solucionar essa questão. Com isso iniciou-se uma tentativa de melhorar a produtividade e escalabilidade, que está diretamente relacionado ao tempo de execução de um programa, como no caso da distribuição de esforços e defeitos por fase e atividade, levando a melhorias gerais tanto no processo como o produto advindo dele. Nesse período percebeu-se que grande parte da fase de teste era basicamente um retrabalho do que já tinha sido feito anteriormente, podendo haver uma redução de custos nesses quesitos.

Posteriormente, foi melhor desenvolvido métodos de seleção para medir a maturidade de software de pessoas e organizações, obtendo uma métrica para avaliar a produtividade e capacidade de identificar bons profissionais. Houve também um grande desenvolvimento das ferramentas de software nesse período, como no caso das ferramentas de teste e gerenciamento, tendo também grandes investimentos em ambientes de suporte. Outro fator de relevância foi a diminuição do tempo de trabalho, dando grande perspectiva de avanço, pois trabalha-se menos e produz mais.

Com o passar do tempo, identificaram grandes problemáticas como softwares que necessitavam de expertise humana, julgamento e colaboração. Uma grande estratégia para enfrentar esse problema é a reutilização de algoritmos que já existiam, para evitar mais programação e um longo período de resposta. A orientação a objetos é um ótimo ponto para ser levado em consideração, principalmente suas estruturas e relações, favorecendo também a modularização. Diversas outras práticas foram de grande relevância, como no caso de analisar erros passados para evitar sua ocorrência no futuro e realizar uma atividade em pouco tempo, pois esse tipo de ação leva a uma menor qualidade do software.

A necessidade de buscar meios para melhorar a produção sempre foi de total relevância, como no caso de treinamentos, melhorias de processos, prototipagem e diversos outros meios. Outra análise necessária é a de buscar soluções que melhorem a situação para um maior quantitativo de pessoas e situações, pois uma solução única na maioria das vezes não tem bom rendimento. A busca por qualidade vai aumentando de acordo com o tempo, principalmente devido aos investimentos, tendo como necessidade um retorno positivo, apresentando uma boa utilidade para as pessoas.

Posteriormente, houve o surgimento da necessidade de uma maior adaptabilidade ao rápido desenvolvimento, sendo necessário uma maior integração de todas as pessoas das equipes nos projetos, para que as necessidades e especificações fossem claras, evitando que o software deixe a desejar em algum quesito. Outro ponto necessário é a possibilidade de ter um plano alternativo, pois questões que não serão resolvidas podem surgir, sendo necessário outro modo de operar, além de buscar boas fontes de informações, para ter um melhor embasamento para o software.

De modo geral, o artigo fornece um ótimo embasamento sobre o desenvolvimento do processo de criação de software, além de ter perspectiva para os anos futuros que condizem muito com a realidade. A partir do desenvolvimento histórico fornecidos, o leitor não só tem acesso a um embasamento histórico dessa área, como obtém diversos conhecimentos que podem ser aplicados nos processos de criação dos próprios programas, evitando erros que já foram cometidos anteriormente. Após a leitura do mesmo, tem-se cada vez mais a noção de que com o passar do tempo, a prioridade na criação de um melhor software tende apenas a aumentar, mostrando um ótimo alinhamento com o que foi fornecido no trabalho analisado.

