

**UNIVERSIDADE SÃO JUDAS TADEU – CAMPUS MOOCA UC –  
USJT-2025.1-GQS-Lab08-Gestão&QualidadeDeSoftware**

**Prof. Calvetti**

**Métricas de Software**

**Alex Aldib – RA: 823165544 Leonardo Ribeiro de Almeida –  
RA: 823213510 Guilherme Figueiredo – RA: 823116166**

# Métricas de Software

## Introdução

O desenvolvimento de software não é apenas sobre codificar, mas também sobre acompanhar, avaliar e melhorar continuamente os processos envolvidos na construção de um sistema. Para isso, utilizam-se as **métricas de software**, que funcionam como instrumentos de medição capazes de fornecer dados objetivos sobre qualidade, produtividade, esforço e desempenho. Neste trabalho, exploramos diferentes abordagens sobre métricas, com base em quatro fontes confiáveis, abrangendo desde práticas do mercado até guias técnicos e governamentais.

## 1. Importância das Métricas no Desenvolvimento de Software

As métricas permitem **avaliar aspectos quantitativos e qualitativos** do software. Elas são fundamentais para:

- Monitorar a **qualidade do código**.
- Controlar prazos e entregas.
- Verificar a satisfação do usuário final.
- Ajudar na **tomada de decisões gerenciais e técnicas**.
- Identificar gargalos no processo de desenvolvimento.

Uma equipe que mede, aprende. E uma equipe que aprende, melhora.

## 2. Métricas Tradicionais e Técnicas Comuns

Segundo a publicação da *ComputerWeekly*, existem dezenas de métricas aplicáveis em diferentes fases do desenvolvimento. Algumas das mais importantes são:

- **Complexidade ciclomática:** mede quantas decisões lógicas existem num trecho de código. Quanto maior a complexidade, mais difícil de testar e manter.
- **Cobertura de testes:** mostra qual parte do código é coberta por testes automatizados.
- **Velocidade da equipe (velocity):** útil em metodologias ágeis, mede quantas tarefas ou pontos são entregues por sprint.
- **Tempo médio de resolução de bugs:** ajuda a identificar eficiência na manutenção.
- **Lead time:** tempo total desde o início de uma tarefa até sua entrega.

Essas métricas são essenciais para encontrar equilíbrio entre **agilidade**, **qualidade** e **eficiência**.

## 3. Métricas no Contexto Ágil

O artigo da Ateliware reforça que nas metodologias ágeis, **medir não é punir**. Pelo contrário, o foco está na **melhoria contínua**, e as métricas servem para **guiar decisões**, **não para vigiar pessoas**. Algumas das mais utilizadas são:

- **Lead time:** mostra quanto tempo se leva, em média, para entregar uma funcionalidade.
- **Throughput:** número de tarefas finalizadas num determinado período.
- **WIP (Work In Progress):** quantidade de tarefas em andamento.
- **Taxa de retrabalho:** quantas vezes uma tarefa precisa ser refeita ou corrigida.

Essas medições ajudam a equipe a se ajustar, descobrir gargalos e otimizar o fluxo de trabalho.

## 4. Métricas no Setor Público e Contratações

Os documentos da *FINEP* e do *Governo Federal (SISP)* tratam das métricas como **instrumentos de controle, planejamento e contratação de serviços de software**. Eles propõem modelos como:

- **Pontos de Função (PF)**: avaliam o tamanho funcional de um sistema baseado no que ele entrega ao usuário.
- **Unidades de Serviço Técnico (UST)**: métrica usada para medir esforço técnico em diferentes tipos de manutenção e evolução de sistemas.
- **Produtividade esperada por perfil**: estimativas de quanto um desenvolvedor deveria entregar em determinado tempo, com base na complexidade do projeto.

Essas métricas são essenciais para contratos de TI com o governo, garantindo **transparência, comparabilidade e controle de custos**.

## 5. Reflexão Crítica

Embora úteis, as métricas devem ser aplicadas com **bom senso**. Usar uma métrica isoladamente pode levar a **interpretações erradas ou decisões injustas**. Por exemplo, exigir alta produtividade sem considerar a complexidade técnica de uma entrega pode prejudicar a qualidade do código. Da mesma forma, medir apenas a velocidade pode incentivar entregas apressadas e com defeitos.

A melhor prática é **combinar diferentes tipos de métricas**, adaptando-as ao contexto da equipe ou da organização, e sempre incentivando uma cultura de aprendizado, e não de punição.

## Conclusão

As métricas de software são aliadas poderosas no desenvolvimento moderno. Elas permitem que equipes entendam melhor seu próprio processo, melhorem continuamente e entreguem valor com mais qualidade. Porém, é essencial lembrar que o objetivo das métricas não é controlar pessoas, mas **gerar conhecimento e orientar decisões**. Seja no setor público, em empresas privadas ou em times ágeis, medir bem é um passo importante para construir software melhor.

## Referências

- ComputerWeekly. *23 métricas de desenvolvimento de software que devem ser monitoradas*.  
Disponível em:  
<https://www.computerweekly.com/br/tip/23-metricas-de-desenvolvimento-de-software-que-devem-ser-monitoradas>
- FINEP. *Guia de Métricas de Software*.  
Disponível em:  
[http://www.finep.gov.br/images/licitacoes/2017/Consulta012017/II\\_GuiaDeMetricasDeSoftware.pdf](http://www.finep.gov.br/images/licitacoes/2017/Consulta012017/II_GuiaDeMetricasDeSoftware.pdf)
- Ateliware. *Métricas ágeis no desenvolvimento de software*.  
Disponível em:  
<https://ateliware.com/blog/metricas-ageis-no-desenvolvimento-de-software>
- Governo Federal. *Roteiro de Métricas do SISP v2.3*.  
Disponível em:  
<https://www.gov.br/governodigital/pt-br/sisp/documentos/arquivos/roteiro-de-metricas-do-sisp-v2-3.pdf>