

Engenharia de Software

Métricas e Indicadores

AGENDA



1. Apresentação

2. Livros

3. Acordo de
Convivência

4. - -

5. - -

6. -

7. -

Apresentação

FORMAÇÃO ACADÊMICA

- ◆ Graduado em Telemática/Telecomunicações - IFCE (2002 - 2008)
- ◆ Especialista em Engenharia de Software - FA7 (2011 - 2013)
- ◆ MSc em Engenharia de Software - UFPE (2011 - 2015)

CURRÍCULO PROFISSIONAL

- ◆ Atuei 4 anos na empresa privada
- ◆ 10 anos no ambiente Público
- ◆ Atualmente Líder Técnico de 45 Projetos de Tecnologia na SEPOG/PMF

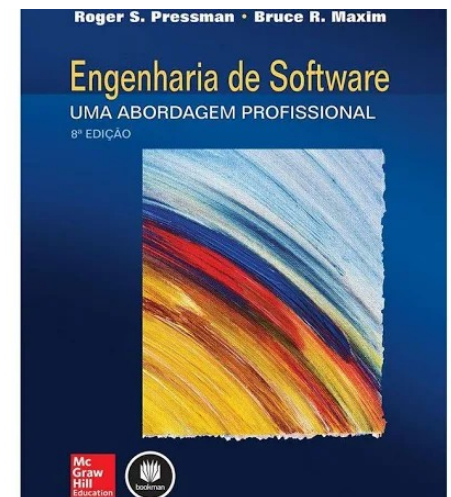
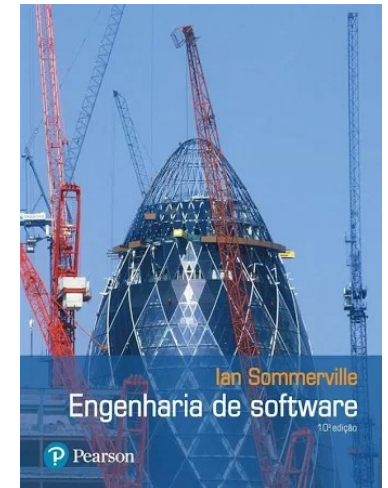
Apresentação

DOCÊNCIA

- ◆ Professor Substituto das Disciplinas de Sistemas de Informação – FA7 (2011 - 2012)
- ◆ Professor da Especialização em Sistemas WEB – FJN (2011 - 2012)
- ◆ Professor de Bancas de graduação em Sistemas de Informações – FA7 (2012)
- ◆ Professor dos Cursos de Tecnologia da Unifanor (2015 – 2018)
- ◆ Professor do Curso de Tecnologia da Unichristus (2018 - Atual)

Livros

- **Engenharia de Software** - 10ª Ed – Ian Sommerville - Pearson
- **Engenharia de Software – Uma abordagem profissional**- 8ª Ed. AMGH



Dicas de Convivência

- ◆ Horários
- ◆ Conversas
- ◆ Dúvidas
- ◆ Celular
- ◆ Avaliações





Questionamentos



Métricas & Indicadores

M e d i d a

Métrica

Medição



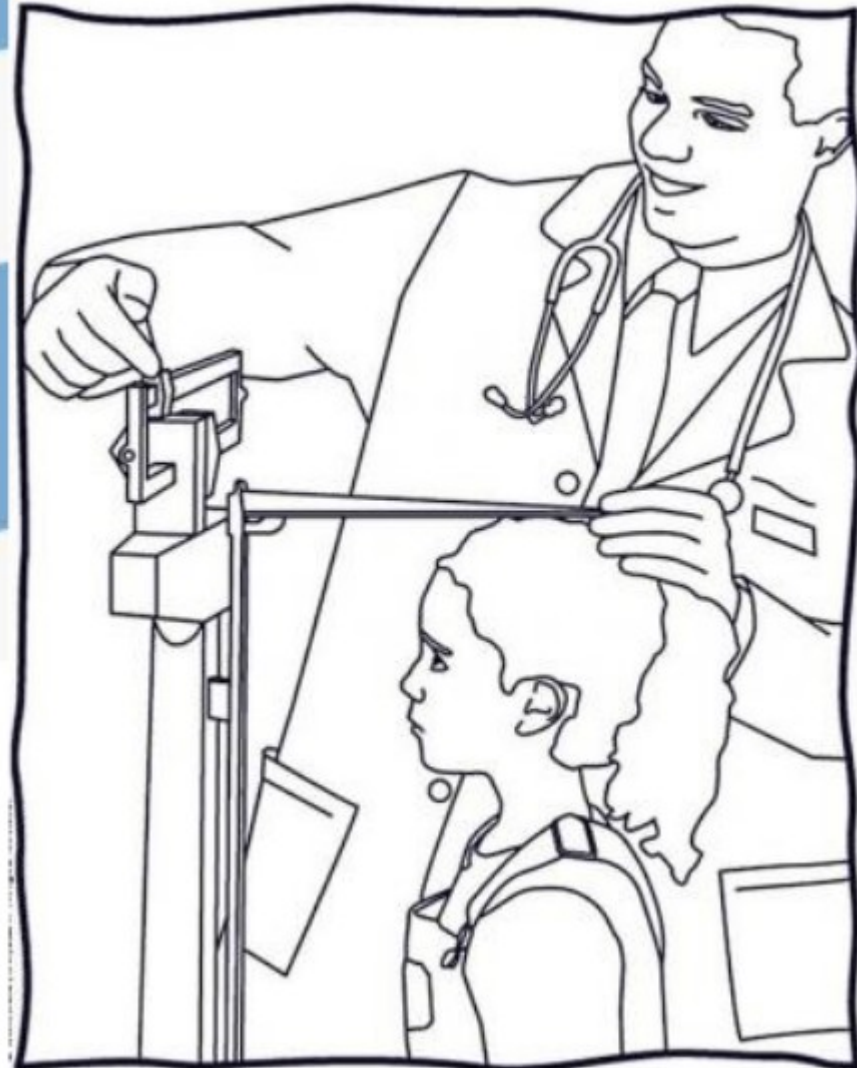
I n d i c a d o r



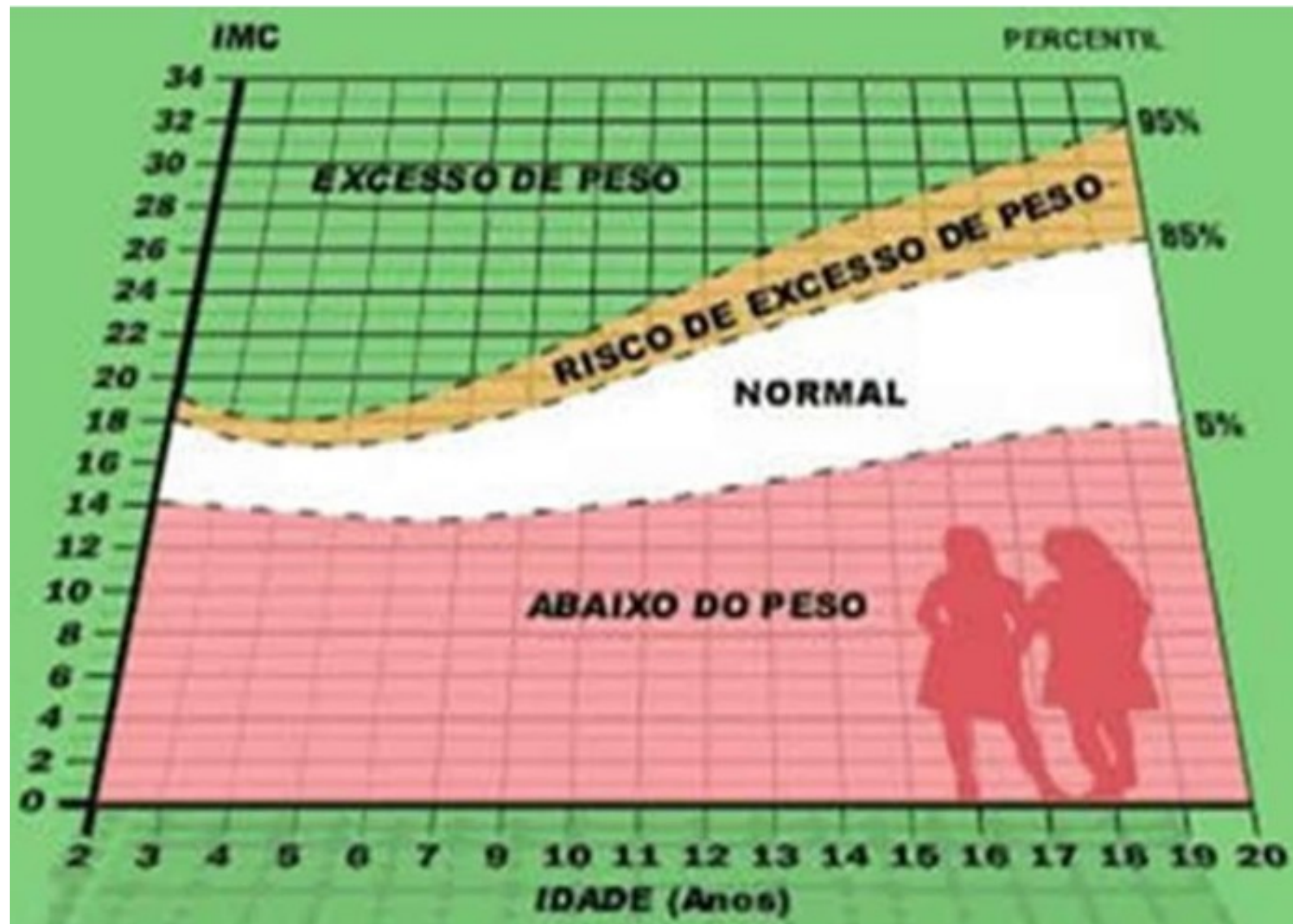
Métricas & Indicadores



$$IMC = \frac{PESO}{ALTURA^2}$$



Métricas & Indicadores



Métricas & Indicadores

Segundo o IEEE, uma **medida** é uma avaliação em relação a um padrão, McGarry diz que é a avaliação de um atributo segundo um método de medição específico, funcionalmente independente de todas as outras medidas e capturando informação sobre um único atributo. Um exemplo de medida é 5cm: centímetro é o padrão e 5 é a medida, que indica quantos múltiplos ou frações do padrão estão sendo representados.

Métricas & Indicadores

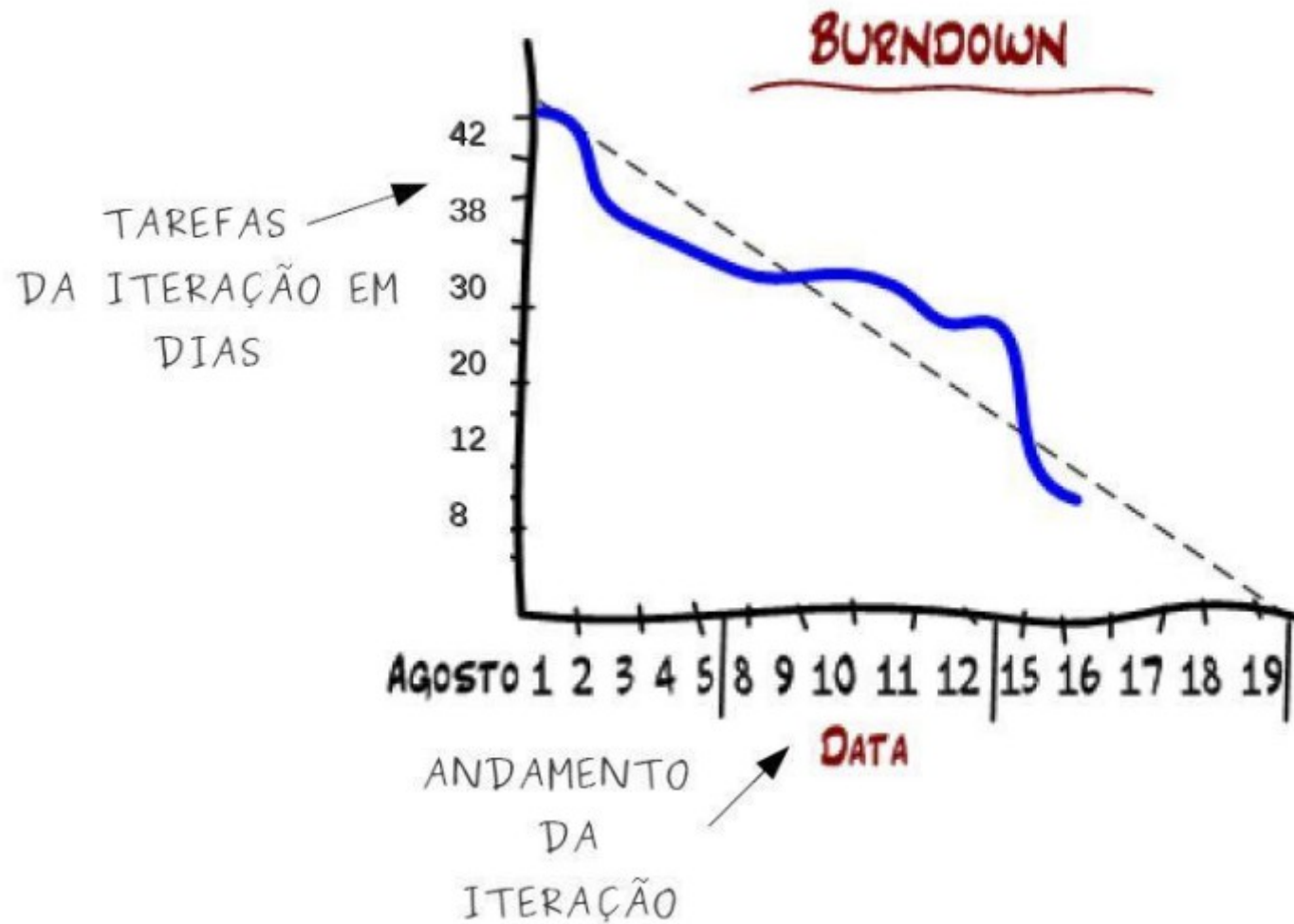
Uma **métrica** é um método para determinar se um sistema, componente ou processo possui um certo atributo. Ela é geralmente calculada ou composta por duas ou mais medidas. Um exemplo de métrica é o número de defeitos encontrados após a implantação: as medidas que compõem essa métrica são o número de defeitos e a fase (ou data) onde o defeito foi identificado.

Métricas & Indicadores

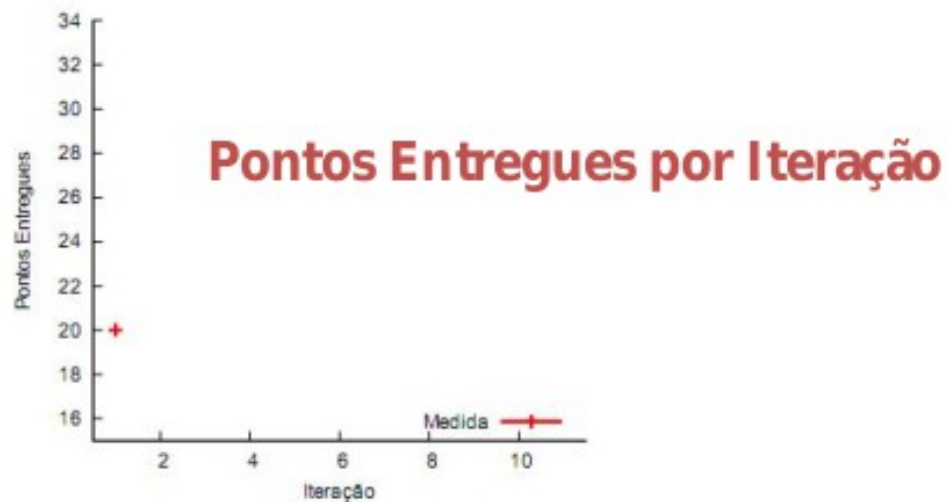
Um **indicador** é um dispositivo ou variável que pode ser configurado para um determinado estado com base no resultado de um processo ou ocorrência de uma determinada condição.

Por exemplo: um semáforo ou uma flag. Conforme a definição do IEEE, um indicador é algo que chama a atenção para uma situação particular

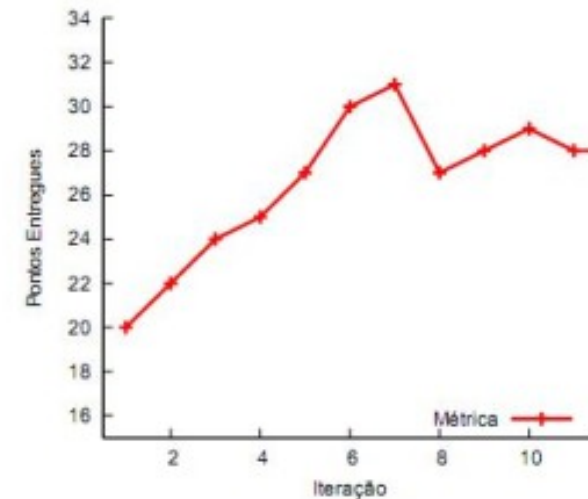
Métricas & Indicadores



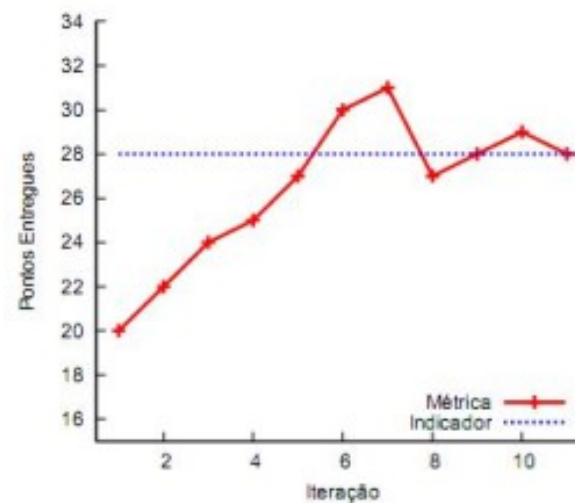
Métricas & Indicadores



(a) Medida



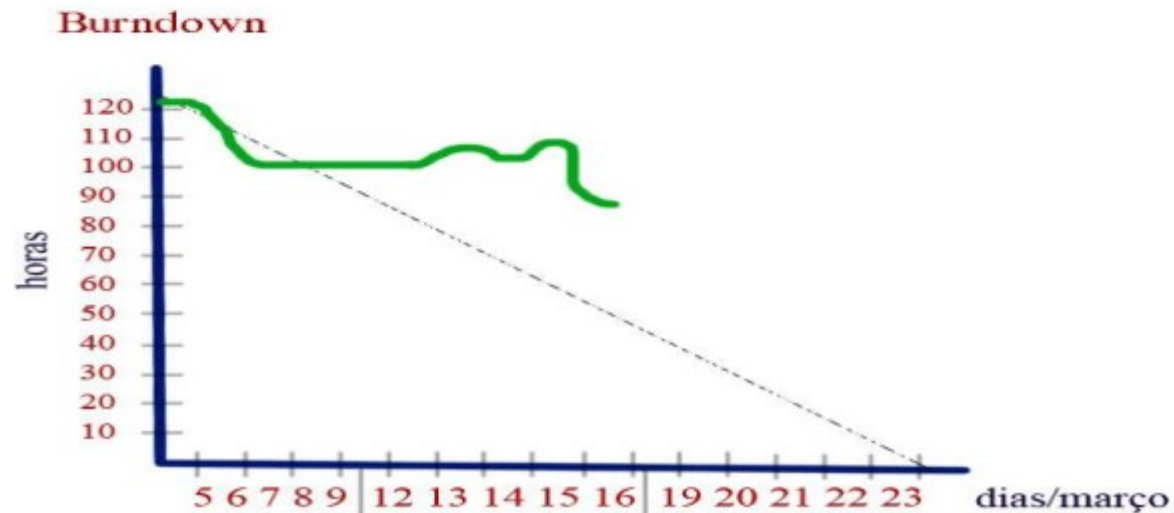
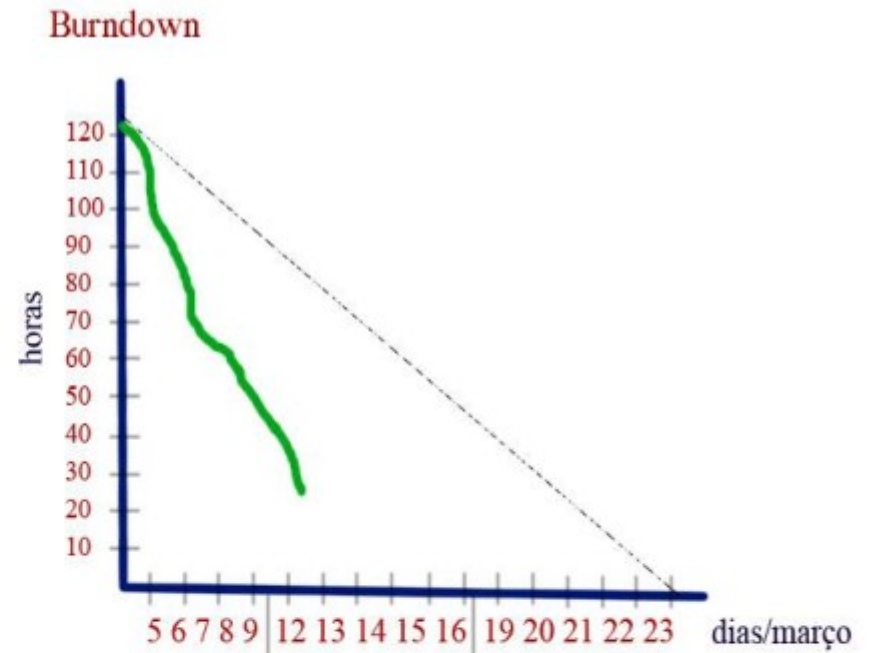
(b) Métrica



(c) Indicador

Fonte: Uso eficaz de métricas em métodos Ágeis de desenvolvimento de software. Danilo Sato, 2007. (Dissertação de Mestrado, USP)

Métricas & Indicadores



Métricas & Indicadores

outros exemplos de métricas ágeis

Task Burndown Chart

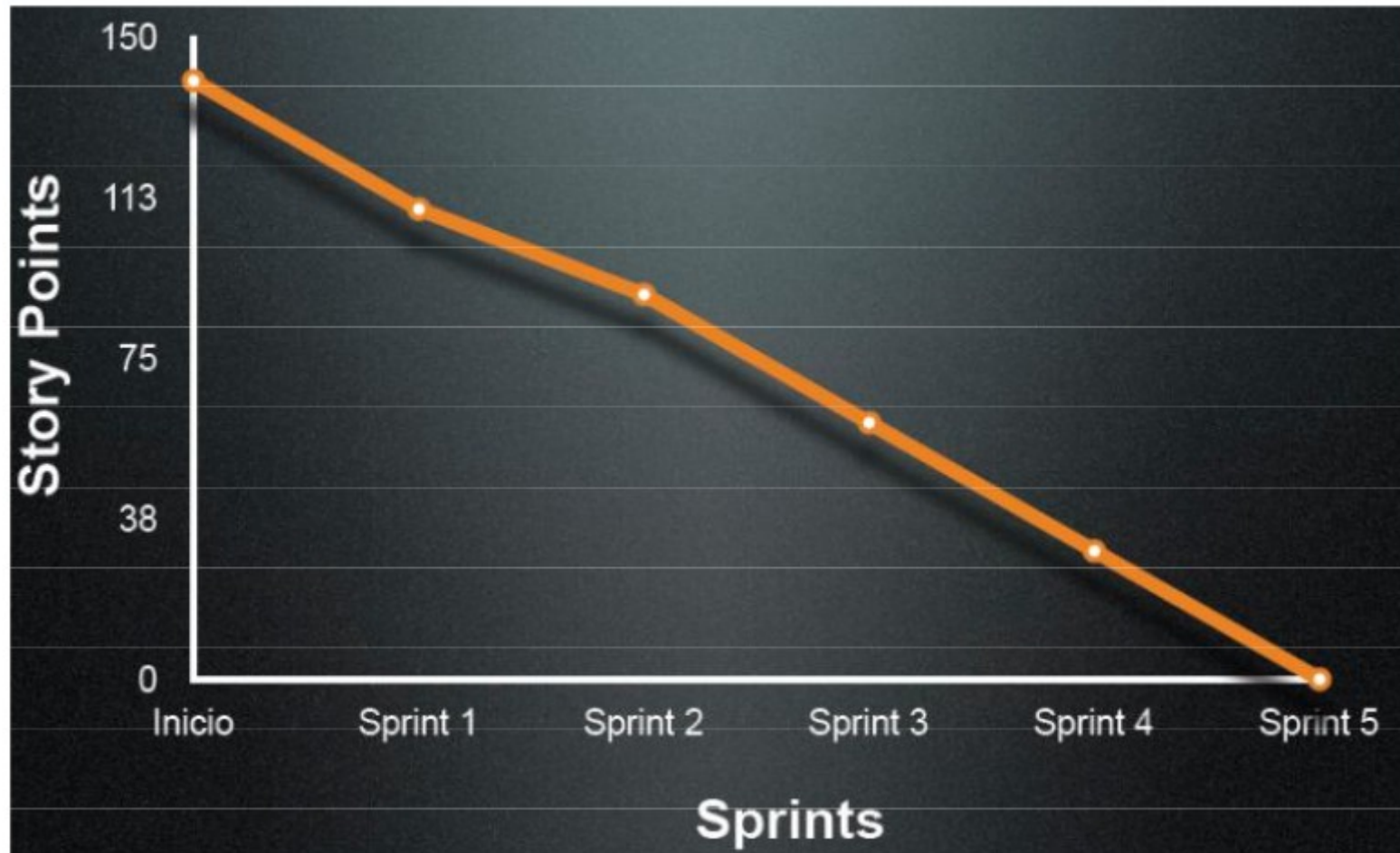


Points Burndown Chart

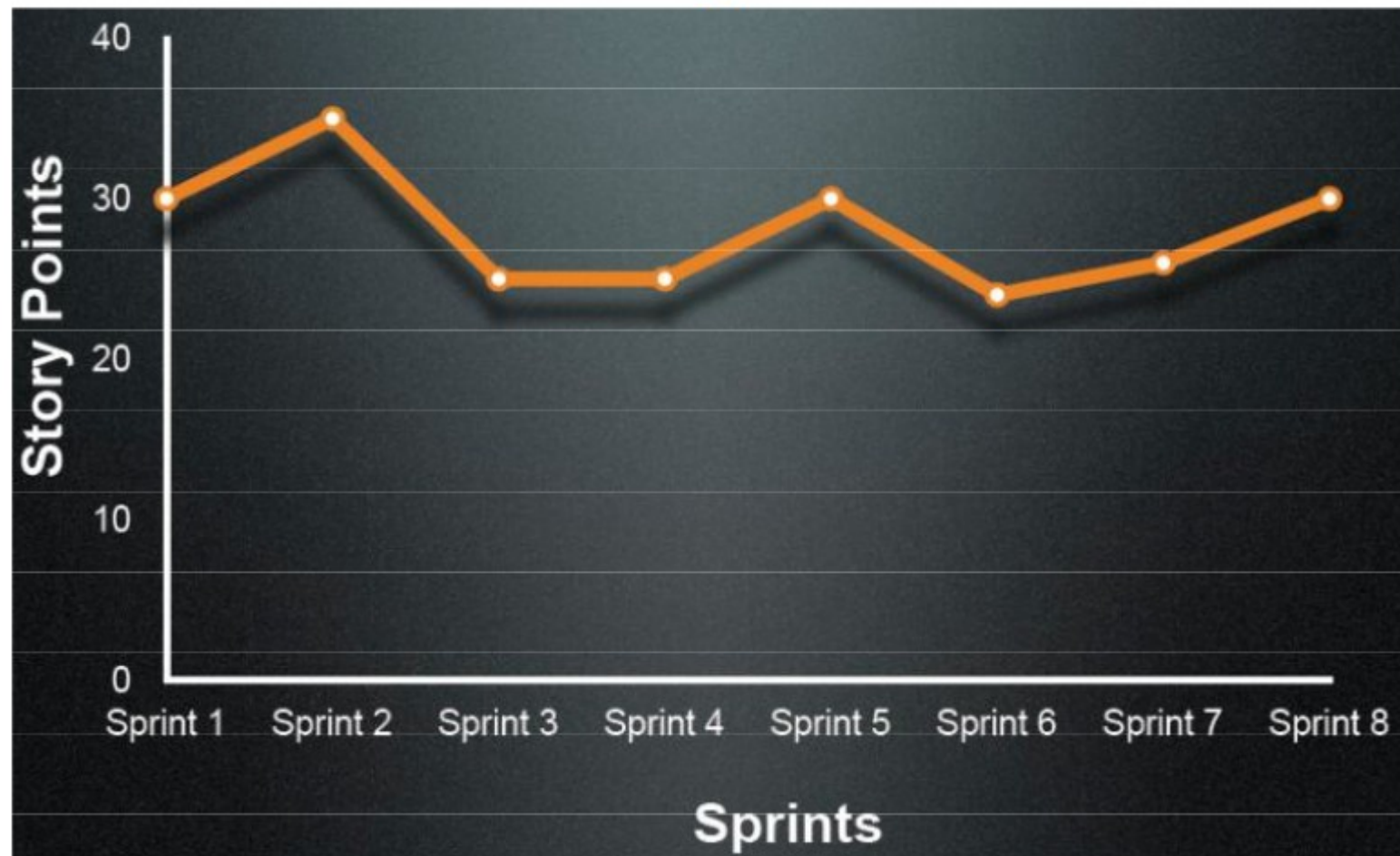


Métricas & Indicadores

Release Burndown Chart

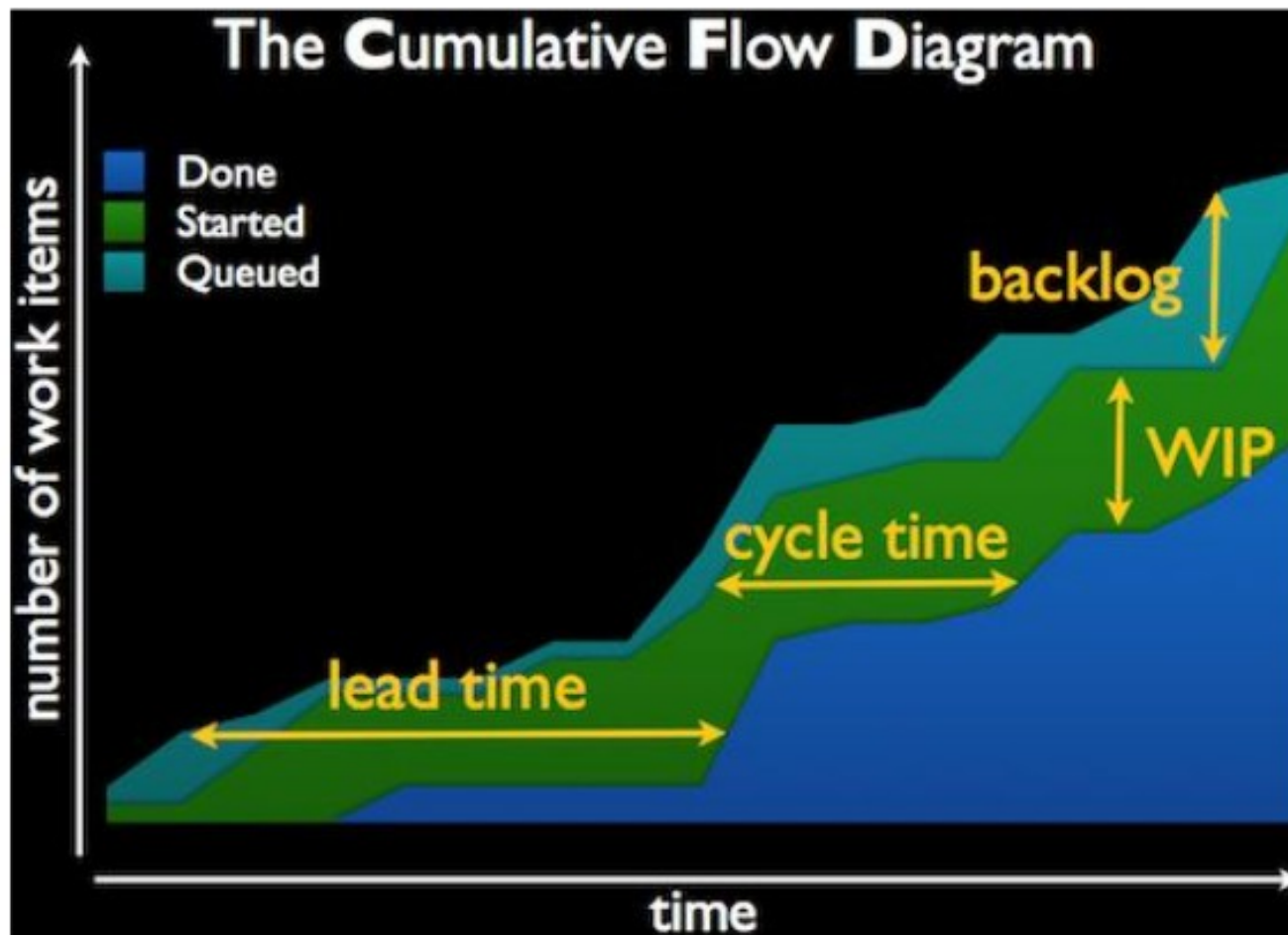


Velocity Chart



Cumulative Flow Diagram

$$\text{WIP} = \text{Throughput} \times \text{Lead time}$$



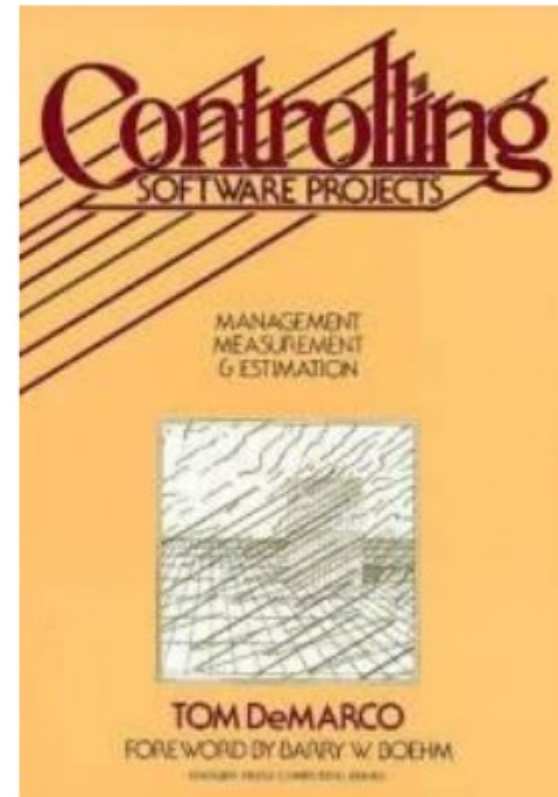
Métricas & Indicadores

mas, é importante medir...?

Métricas & Indicadores

Tom DeMarco, 1982

“You can’t
control what
you can’t
measure”



Tom DeMarco, 2009

27 anos depois...

Tom DeMarco mudou de ideia em
relação ao controle?

Métricas & Indicadores

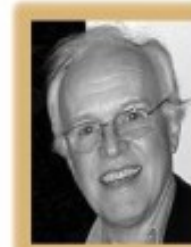
Tom DeMarco, 2009

“Desenvolvimento de software é e sempre será de alguma maneira experimental. Quanto mais você foca em controle, maior a probabilidade de seu projeto estar entregando algo de valor baixo”

Software Engineering: An Idea Whose Time Has Come and Gone?

Tom DeMarco

We're now just past the 40th anniversary of the NATO Conference on Software Engineering in Garmisch, Germany, where the discipline of software engineering was first proposed. Because some of my early work became part of that new discipline, this seems like an appropriate moment for reassessment.



My early metrics book, *Controlling Software Projects: Management, Measurement, and Estimation* (Prentice Hall/Yourdon Press, 1982), played a role in the way many budding software engineers quantified work and planned their projects. In my reflective mood, I'm wondering, was its advice correct at the time, is it still relevant, and do I

Compelled to Control

The book's most quoted line is its first sentence: "You can't control what you can't measure." This line contains a real truth, but I've become increasingly uncomfortable with my use of it. Implicit in the quote (and indeed in the book's title) is that control is an important aspect, maybe the most important, of any software project. But it isn't. Many projects have proceeded without much control but managed to produce wonderful products such as GoogleEarth or Wikipedia.

To understand control's real role, you need to distinguish between two drastically different kinds of projects:

- Project A will eventually cost about a million dollars and produce value of around \$1.1 million.
- Project B will eventually cost about a million

Métricas & Indicadores

Tom DeMarco, 2009

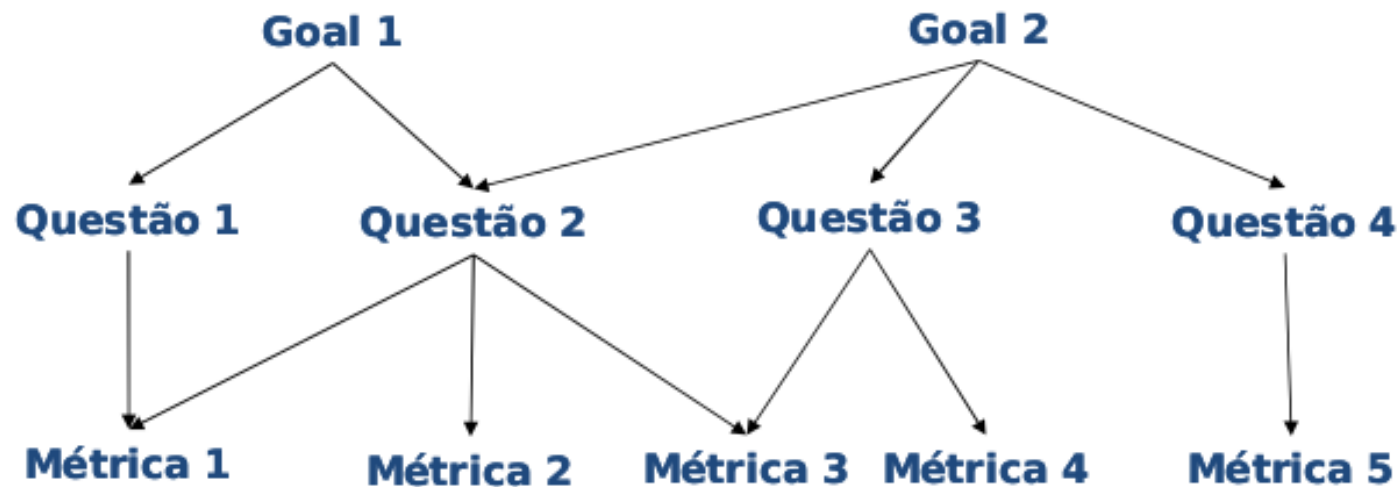
“... então, como você gerencia um projeto que não pode **controlar**?”

“... você **gerencia as pessoas e controla o tempo e o custo**. Estou sugerindo um approach de gestão muito próximo de **métodos ágeis**.”

O Paradigma Goal Question Metrics (GQM)

- Usado para definir o conjunto de métricas a ser coletado;
- Proposto por:
 - Basili and Rombach's, Goal-Question-Metrics Paradigm, IEEE Transactions on Software Engineering, 1988.
- Baseia-se no fato de que deve existir uma necessidade clara associada a cada métrica.

O Paradigma Goal Question Metrics (GQM)



O Paradigma Goal Question Metrics (GQM)

- Inicia-se com a identificação dos interessados na medição;
- Com base nos interessados, estabelecem-se os principais objetivos da medição para a organização, o projeto ou uma tarefa específica.
 - Ex: reduzir defeitos, aumentar produtividade, etc.
- A partir dos objetivos, geram-se perguntas cujas respostas dirão se os objetivos foram ou não alcançados
 - Ex: Qual a taxa de defeito atual? Qual a taxa de defeito após a implantação do novo processo?
- A partir das perguntas, definem-se métricas: que dados serão necessários? Quais os formatos? Como coletar (fórmula e processo)? Onde armazenar e como utilizar?

Exemplo do uso do GQM

- Objetivo: Assegurar que todos os defeitos são corrigidos antes do software ser liberado para uso.
- Perguntas:
 - Quantos defeitos temos atualmente?
 - Qual o status de cada defeito?
 - Qual a cobertura dos testes?
- Métricas:
 - Número de defeitos;
 - Número de defeitos por status;
 - Número de casos de testes planejados x executados;
 - Número de requisitos testados.

Definição de Métricas: Método GQM

- GQM - *Goal-Question-Metric* (Objetivo-Questão-Métrica);
- Passos:
 - Listar os objetivos da organização ou projeto;
 - A partir de cada objetivo, derivar questões que devem ser respondidas para determinar se os objetivos foram atingidos;
 - Decidir o que deve ser medido, de forma a ser capaz de responder às questões de forma apropriada.

GQM - Exemplo

- Objetivo: Melhorar a qualidade dos processos de verificação e validação.
- Questões:
 - Qual a quantidade de erros encontrados na revisão dos artefatos?
 - Qual a quantidade de erros encontrados nos testes?
 - Qual a quantidade de erros detectados quando o sistema já está em operação?
- Métricas:
 - Defeitos encontrados nas revisões / PF (Ponto de Função);
 - Defeitos encontrados nos testes / PF;
 - Defeitos encontrados na operação / PF.

Propriedades desejáveis em uma métrica

- Facilmente calculada, entendida e testada;
- Passível de estudos estatísticos;
- Expressa em alguma unidade;
- Obtida o mais cedo possível no ciclo de vida do software;
- Passível de automação;
- Repetível e independente do observador;
- Sugere uma estratégia de melhoria.

Produtividade medida em Linhas de Código?!?!

-2000 Lines Of Code

Author: Andy Hertzfeld

Date: February 1982

Characters: Bill Atkinson

Summary: It's hard to measure progress by lines of code

In early 1982, the Lisa software team was trying to buckle down for the big push to ship the software within the next six months. Some of the managers decided that it would be a good idea to track the progress of each individual engineer in terms of the amount of code that they wrote from week to week. They devised a form that each engineer was required to submit every Friday, which included a field for the number of lines of code that were written that week.

Bill Atkinson, the author of Quickdraw and the main user interface designer, who was by far the most important Lisa implementor, thought that lines of code was a silly measure of software productivity. He thought his goal was to write as small and fast a program as possible, and that the lines of code metric only encouraged writing sloppy, bloated, broken code.

He recently was working on optimizing Quickdraw's region calculation machinery, and had completely rewritten the region engine using a simpler, more general algorithm which, after some tweaking, made region operations almost six times faster. As a by-product, the rewrite also saved around 2,000 lines of code.

He was just putting the finishing touches on the optimization when it was time to fill out the management form for the first time. When he got to the lines of code part, he thought about it for a second, and then wrote in the number: -2000.

I'm not sure how the managers reacted to that, but I do know that after a couple more weeks, they stopped asking Bill to fill out the form, and he gladly complied.

Possíveis problemas com métricas

- Ex: Comparar a produtividade de engenheiros em termos de linha de código
 - Está sendo utilizado a mesma unidade de medida?
 - O que é uma linha de código válida?
 - O contexto considerado é o mesmo?
 - Todos os engenheiros são familiarizados com a linguagem de programação?
 - O que se quer realmente é o tamanho do código?
 - E a qualidade do código?
 - Como o resultado será interpretado?
 - Produtividade média de um engenheiro?
 - O que se quer com o resultado?
 - Comparar a produtividade do processo de software?

Métricas & Indicadores



Métricas & Indicadores

**“nossa
produtividade é
de 60%”**

**“não tenho
interesse em
medir
produtividade”**

**“podemos ser
produtivos, mas
não gerarmos
resultados”**

**“a produtividade
da minha equipe
é acompanhada
através do
esforço
realizado”**

- se o conceito sobre produtividade não é muito claro, o que podemos dizer sobre a medição e melhoria da produtividade ?

Métricas & Indicadores

**Produtividade:
O que é exatamente isso?**

Métricas & Indicadores

“Productivity is the amount of output created per unit input used”

wikipedia

“Productivity is a measure relating a quantity and/or quality of output to the inputs required to produce it”

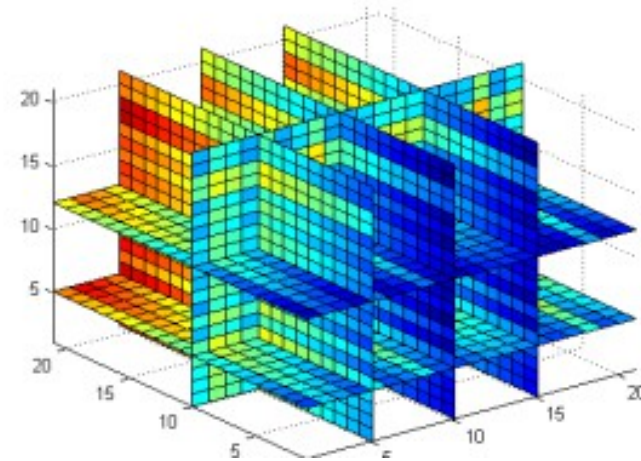
Cambridge Dictionary

$$\text{Produtividade} = \frac{\text{OUTPUT}}{\text{INPUT}}$$

Métricas & Indicadores

$$\text{Produtividade} = \frac{\text{Valor Produzido}}{\text{Valor Consumido}}$$

o valor produzido
deve ser analisado
em múltiplas
dimensões!



Métricas & Indicadores

Valor Produzido

- Funcional?
- Financeiro?
- Satisfação do cliente?
- Satisfação da equipe?
- Inovação?
- Visibilidade?
- Experiência?
- ...

Valor Consumido

- Esforço
- Financeiro
- ...

Métricas & Indicadores

Existem muitas medidas de produtividade na literatura, mas várias são baseadas em quantidade, e não em valor!

- LOC
- Ponto de Função
- COSMIC-FFP
- MKII FP
- Ponto de Caso de Uso
- Feature Points

**Mas como medir se
ainda não sabemos
o que deve ser considerado
como valor?**

Database Code in Table 1	Reference	No. Programs	Environment	Geographical Scope	Productivity Measure
ESA Data	Our research		Space/Military	Europe	L.O.C.
A	Al, 1976		Large	USA	L.O.C.
B	Al, 1982			USA	F.P.
C	Bailey, 1982				L.O.C.
D	Behren, 1982				F.P. and L.O.C.
E	Behren, 1982				L.O.C.
F	Behren, 1982				L.O.C.
G	Behren, 1982				L.O.C.
H	Behren, 1982				L.O.C.
I	Behren, 1982				L.O.C.
J	Behren, 1982				L.O.C.
K	Behren, 1982				L.O.C.
L	Behren, 1982				L.O.C.
M	Behren, 1982				L.O.C.
N	Behren, 1982				L.O.C.
O	Behren, 1982				L.O.C.
P	Behren, 1982				L.O.C.
Q	Vesil, et al. 1984	44	Mixture (from ITT)	19 countries	L.O.C.

Métricas & Indicadores

Definindo o significado de **Valor Produzido para a organização (e para cada projeto).**

Métricas & Indicadores

