



MeasureSoftGram: uma proposta de observação multidimensional da qualidade de produto de software em ambientes de desenvolvimento e experimentação contínua

Exame de Qualificação de Doutorado

Hilmer Rodrigues Neri

Orientador: Guilherme Horta Travassos

Agenda

Introdução

- motivação, problema, objetivo, questão pesquisa e metodologia

Revisão da Literatura

- Medição, Medidas e Métricas
- Modelos de Qualidade de Produto de Software
- Desenvolvimento e Experimentação Contínua de Software
- Software *Analytics*
- Espaço Tensorial
- Contratações Públicas Brasileiras de Desenv. Software

MeasureSofGram: a proposta

Estudo de Viabilidade

Ameaças à Validade da Pesquisa

Conclusões Preliminares e Próximos passos

Contextualização

Desenvolvimento de produtos e Gerenciamento de projetos de software

- monitoramento da execução e garantia da qualidade

Verificação e Validação

Qualidade de Software

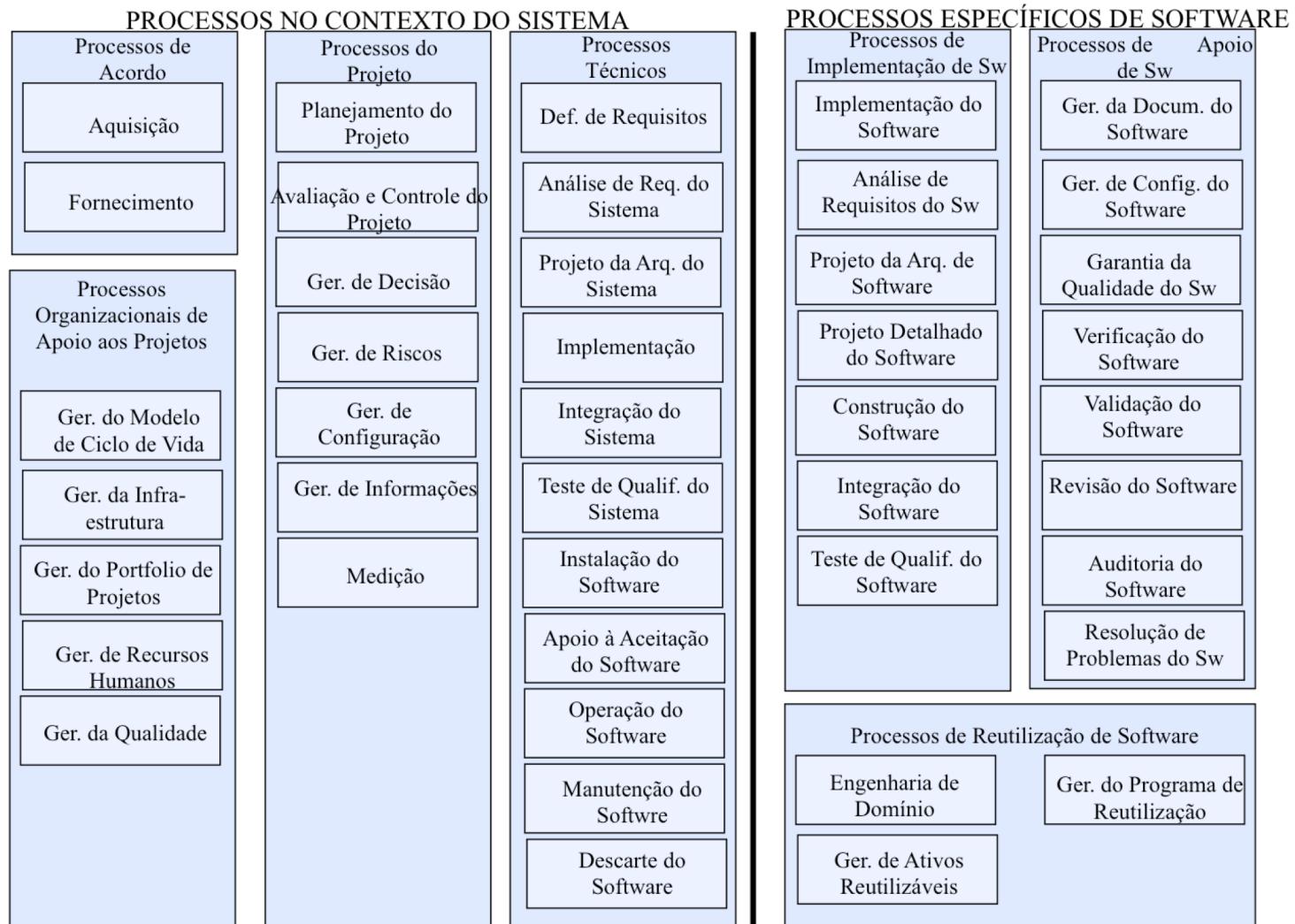
- Produto

- Família ISO 25000/ (*Software Product Quality Requirements and Evaluation*)

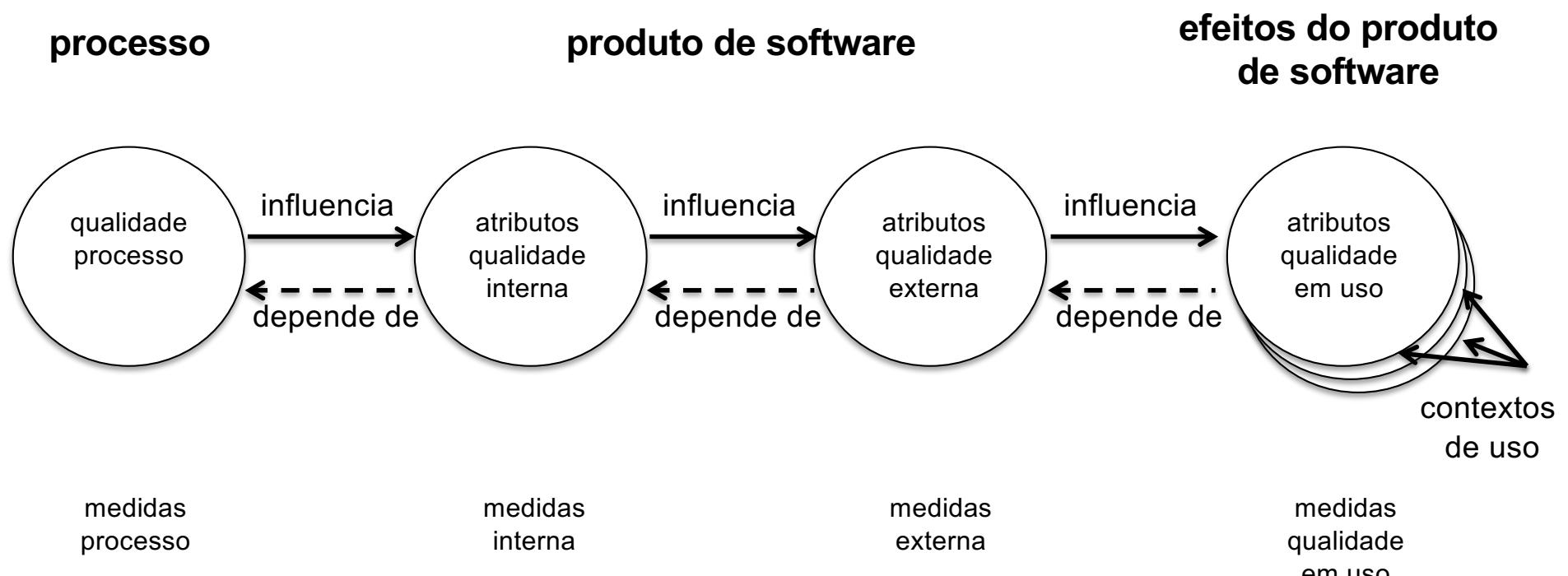
- Processo

- ISO 15939 (*Systems and Software Engineering – Measurement Process*)
 - ISO 12207 (*Software Life Cycle Process*)
 - ISO 15504 (Information Technology – Process Assessment)

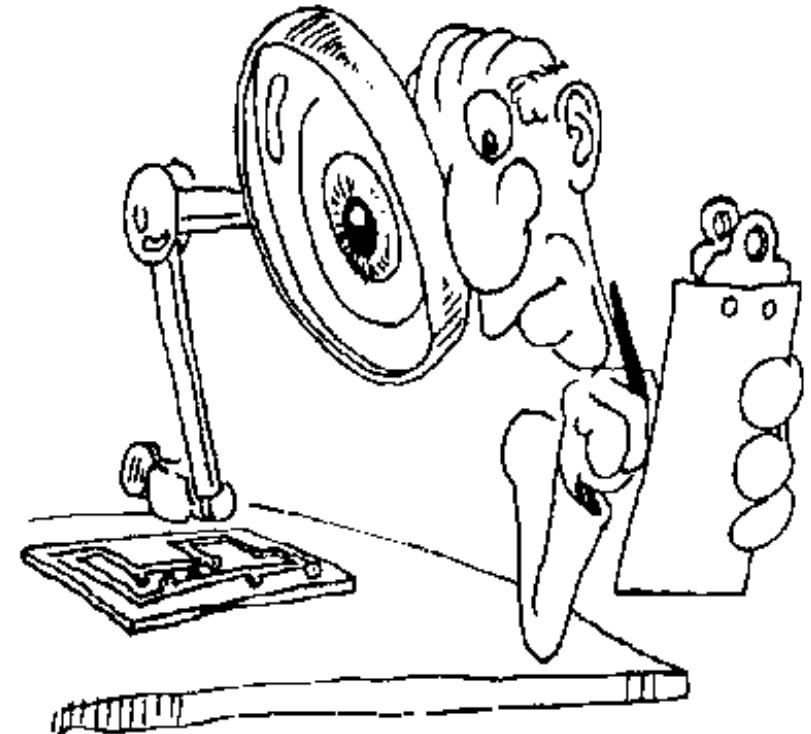
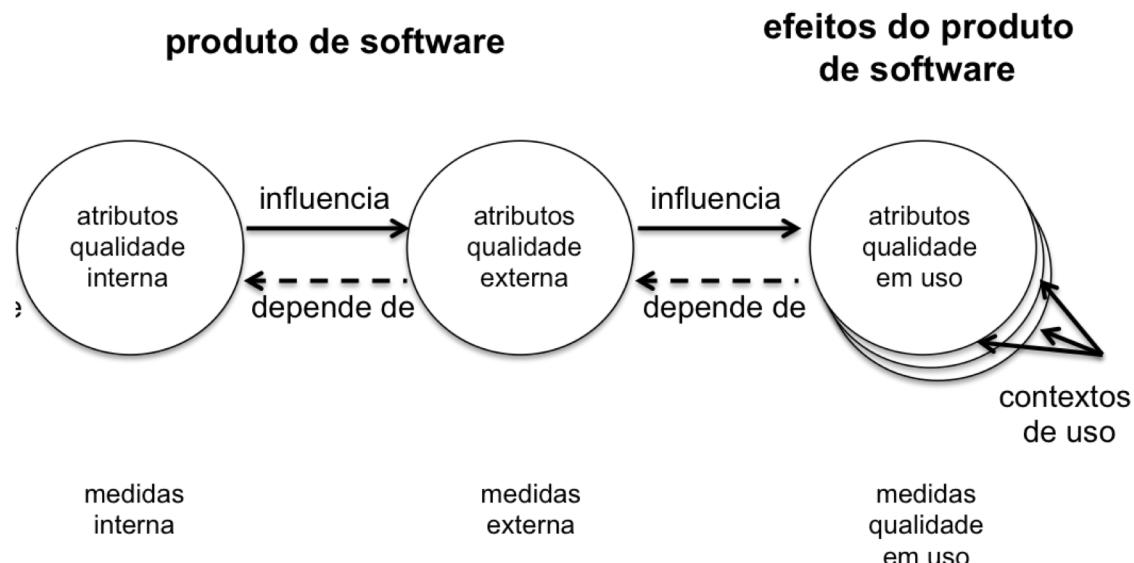
Contextualização



Contextualização



Foco desta Pesquisa



Introdução

Motivação

- Qualidade de Produto:
 - objeto de estudo da eng. software há mais de 4 décadas McCall(1977) (López et al., 2018)
 - comumente observada de forma unidimensional (linear) e (ISO/IEC 25010 2010) (Wagner, 2012)
- Limitado conhecimento acerca das relações entre os fatores que descrevem a qualidade de produto
 - usabilidade X adequação funcional (Henningsson e Wohlin, 2002)
 - qualidade produto ISSO 25010 (Haoues, 2017)
- Analisar dados oriundos da produção do software, em conjunto, de forma a apoiar decisões técnicas, gerenciais e do negócio, não é trivial.
 - automação é determinante para sucesso na medição da qualidade de produto (Boehm, 1978)
 - dificuldade em agregar estudos (Henningsson, 2002)
 - a análise deve considerar informações extraídas de diferentes artefatos, de forma a apoiar a análise e tomada de decisão (Abdellatif et al., 2015)

Introdução

Motivação

- Estudos experimentais sobre medição e medidas de qualidade de software geralmente relatam:
 - falta de rigor no processo de medição e definição de medidas (Kitchenham, 2010);
 - problemas em escalas entre medidas (Juristo e Moreno, 2010);
 - as características do contexto que dificultam, ou mesmo impossibilitam, a comparação de medidas entre diferentes produtos ou projetos (Sjoberg, 2012)
 - a dificuldade de estabelecer valores de referência para medidas de software (Lavazza, 2016).
- A forma de se desenvolver software mudou!
 - Engenharia de Software Contínua (Fitzgerald e Stol, 2017)
 - Experimentação Contínua (Schermann, 2018)

O Problema

A falta de compreensão ou clareza acerca das características do produto de software, além das suas relações, afetam a aceitação de releases. Isso acarreta em desperdício de recursos e contribui para tomada de decisões técnicas e gerenciais não assertivas(ruins), comprometendo:

- I. a qualidade do produto de software em desenvolvimento/operação;
- II. o comportamento do software em uso e;
- III. a estratégia de negócios das organizações.

O Problema

A observação da qualidade de produto de software de forma unidimensional (linear), impede a percepção das relações entre suas características e subcaracterísticas.

Portanto,

os efeitos das relações e também os colaterais, são desconhecidos e isso afeta a qualidade e por conseguinte compromete a tomada de decisão.

Objetivo

Analisar

a qualidade de sistemas de software web (produto e uso) em ambientes de engenharia de soft. contínua

**com o
propósito de**

caracterizar fatores e medidas da qualidade, além das suas relações, a fim de prover, a partir de evidências, suporte a tomada de decisão técnico-gerencial sobre a aceitação de releases

com respeito à

eficiência

do ponto de vista pesquisador

no contexto

de contratações públicas brasileiras federais de serviço de desenvolvimento de software

Questão de Pesquisa

“

Como observar a qualidade de produto de software, de forma multidimensional, em ambientes de desenvolvimento e experimentação contínua, para efeito da aceitação de releases na contratação pública brasileira de desenvolvimento de software?

”

Metodologia



Metodologia



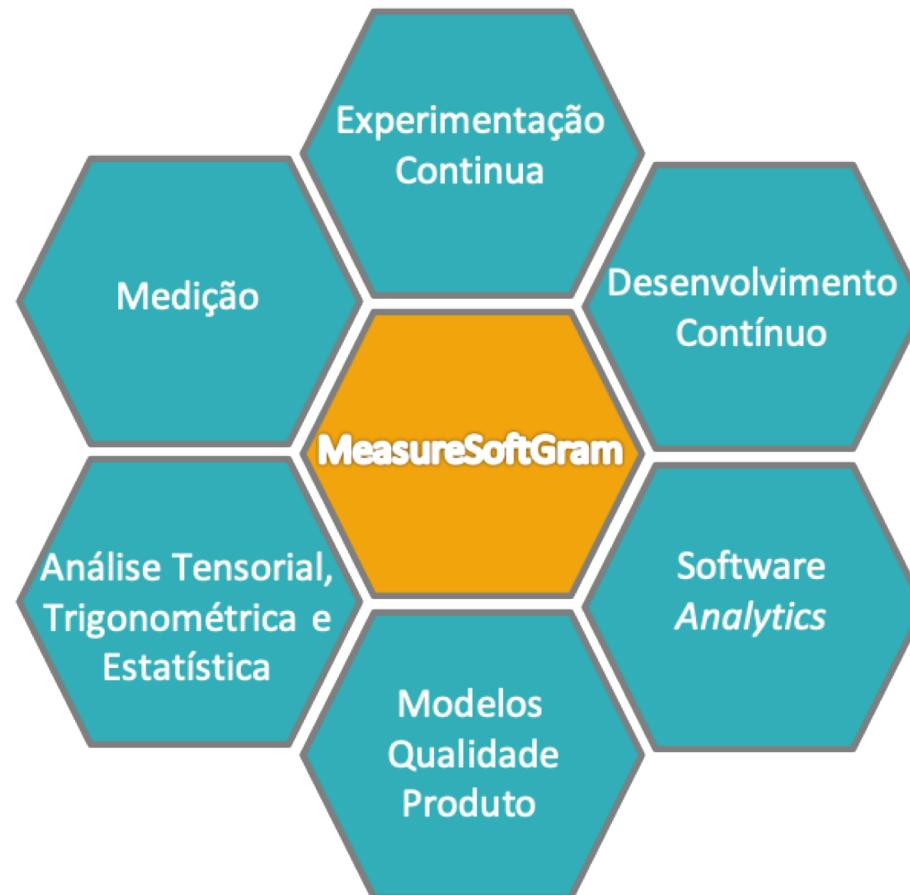
Autor(a)	Título	Ano	Fonte	Relação com esta pesquisa
Kitchenham, B.	What's up with software metrics?--A preliminary mapping study	2010	Journal of systems and software	métricas e medidas
Tahir A; MacDonell, S. G	A systematic mapping study on dynamic metrics and software quality	2012	IEEE International Conference on Software Maintenance	estáticas de software
Migue J. P. ; Mauricio D; Rodriguez G.	A Review of Software Quality Models for the Evaluation of Software Products	2014	International Journal of Software Engineering & Applications International Workshop on Software Measurement and the International Conference on Software Process and Product Measurement	métricas e medidas dinâmicas de software
Ouhbi, S.; Idri, A.; Alemán, J L F.; Toval, A.	Evaluating Software Product Quality: A Systematic Mapping Study	2014	International Conference on Software Process and Product Measurement	modelos de qualidade de produto de software
Abdellatif T. M.; Capretz L. F.; Ho D.	Software Analytics to Software Practice: A Systematic Literature Review	2015	International Workshop on BIG Data Software Engineering	modelos de qualidade de produto de software
Papalexakis, E. E.; Faloutsos C.; Sidiropoulos N. D.	Tensors for Data Mining and Data Fusion: Models, Applications, and Scalable Algorithms	2016	ACM Transactions on Intelligent Systems and Technology	análise sistemática de software
Fitzgerald B.; Stol Klaas-Jan	Continuous software engineering: A roadmap and agenda	2017	The Journal of Systems & Software	aplicação de tensores na área de computação
Schermann G; Cito J.; Leitner P.; Zdun U; Gall H.C.	We're Doing It Live: A Multi-Method Empirical Study on Continuous Experimentation	2018	Information and Software Technology (IST)	desenvolvimento contínuo
				experimentação contínua

Metodologia



Normativo	Descrição	Ano	Responsável	Endereço de Acesso
Lei nº 8.666	Institui normas para licitações e contratos na Administração Pública	1993	Presidência da República	http://www.planalto.gov.br/ccivil_03/leis/L8666cons.htm
Acórdão nº 2.471	Instrui a criação da antiga Secretaria de Logística de Tecnologia da Informação-SLTI/MP, do então Ministério do Planejamento, Orçamento e Gestão-MP	2008	Tribunal de Contas da União	https://portal.tcu.gov.br/biblioteca-digital/normativo-acordao-2471-2008-plenario-tcu.htm
Instrução Normativa nº 04	Dispõe sobre o processo de contratação de soluções de tecnologia da informação pelos órgãos integrantes do Sistema de Administração dos Recursos de Informação e Informática (SISP) do Poder Executivo Federal	2010 (atualizada em 2014)	Antiga Secretaria de Logística de Tecnologia da Informação-SLTI/MP	https://www.governodigital.gov.br/documentos-e-arquivos/legislacao/IN%20SLTI%20MP%2004%202010%20-%20Consolidada%20-%20Modificada%20%20pela%20I.pdf/view
Acórdão 2.314	Sinaliza, ponderando os riscos associados, sobre a contratação de serviços de desenvolvimento de software utilizando metodologias ágeis (não altera o normativo vigente)	2013	Tribunal de Contas da União	https://contas.tcu.gov.br/etcu/ObterDocumentoSisdoc?seAbrirDocNoBrowser=true&codArqCatalogado=6488150
Guia de projetos com práticas de métodos ágeis para o SISP	Oferrece orientações e diretrizes para contratação de serviço de desenvolvimento de software utilizando práticas ágeis	2015	SISP	http://www.sisp.gov.br/guiaagil/wiki/download/file/Guia_de_Projetos_%C3%81geis
Roteiro de métricas de software do SISP_v2.2	Em essência apresenta um roteiro de métricas com base nas regras de contagem de pontos por função, como a principal medida utilizada em contratos públicos de desenvolvimento de software	2016	SISP	https://www.governodigital.gov.br/documentos-e-arquivos/Roteiro%20de%20Metricas%20de%20Software%20do%20SISP%20-%20v2.0.pdf
Processo de Software do IN04/MP/2010 SISP	Em essência é o desenho do fluxo do processo previsto na IN04/MP/2010	-	SISP	http://www.sisp.gov.br/pwsisp/default

Áreas de Conhecimento

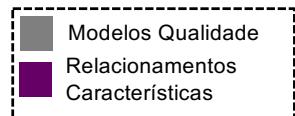


Revisão da Literatura



Modelos de Qualidade ao Longo do Tempo

e os estudos sobre os relacionamentos entre características e subcaracterísticas



1970-90

McCall (77)
Boehm (78)
Rocha (83)
Boehm (86)
FURPS (92)
Dromey (95)

2000

ISO 9126 (01)
Bertoa (02)
Hennigsson (02)
Georgiadou (03)
CapGemini (03)
Rawashdel (06)
OpenBRR (06)

Andreu (07)
ISO 25000 (08)
SQO-OSS (08)
Svahnberg (09)
QualOSS (09)
Squale (09)

2010-18

Alvaro (10)
Upadhyay(11)
Al-Badareen (12)
Aldaajeh (12)
Quamoco (12)

Miguel J. (14)
Qatch (17)
Haoues (17)
Q-Rapids (18)

Fatores da Qualidade de Produto

Characteristic	McCall	Boehm	FUR PS	Dromey	ISO-9126	ISO-25010
Accuracy					X	X
Adaptability			X			X
Analyzability					X	X
Attractiveness					X	X
Changeability					X	X
Correctness	X					X
Efficiency	X	X		X	X	X
Flexibility	X					
Functionality			X	X	X	X
Human Engineering		X				
Installability					X	X
Integrity	X					X
Interoperability	X					X
Maintainability	X			X	X	X
Maturity					X	X
Modifiability						X
Operability					X	X
Performance			X		X	X
Portability	X	X		X	X	X
Reliability	X	X	X	X	X	X
Resource utilization					X	X
Reusability	X			X		X
Stability					X	X
Suitability					X	X
Supportability			X		X	X
Testability	X	X			X	X
Transferability						X
Understandability		X			X	X
Usability	X		X	X	X	X

ISO 25020

- A ISO 9126 e sua atualização, a ISO 25010, tem sido o modelo de qualidade de produto mais estudado ao longo do tempo (Ouhb et al, 2014)

Qualidade De Produto							
Adequação Funcional	Eficiência de Desempenho	Compatibilidade	Usabilidade	Confiabilidade	Segurança	Facilidade de Manutenção	Portabilidade
Integridade Funcional	Comportamento Temporal	Coexistência	Adequação no Reconhecimento	Maturidade	Confidencialidade	Modularidade	Adaptabilidade
Corretude Funcional	Utilização de Recursos	Interoperabilidade	Capacidade de Aprendizagem	Disponibilidade	Integridade	Reutilização	Facilidade de Instalação
Pertinência Funcional	Capacidade		Proteção Contra Erros do Uso	Tolerância a Falhas	Não Repudiação	Facilidade de Análise	Capacidade Substituição
			Operacionalidade	Capacidade de Recuperação	Responsabilização	Mutabilidade	
			Acessibilidade		Autenticidade	Testabilidade	
			Estética de Interface				

Qualidade em Uso				
Eficiência	Eficácia	Satisfação	Liberdade do risco	Cobertura do Contexto
Eficiência	Eficácia	Facilidade de Uso	Mitigação de risco econômico	completude do contexto
		Confiança	Mitigação de riscos à saúde e segurança	Flexibilidade
		Prazer	Mitigação de riscos de ambiente	
		Conforto		

Características e Sub-Características Qualidade de Sistemas (Produto e em Uso)

Revisão da Literatura



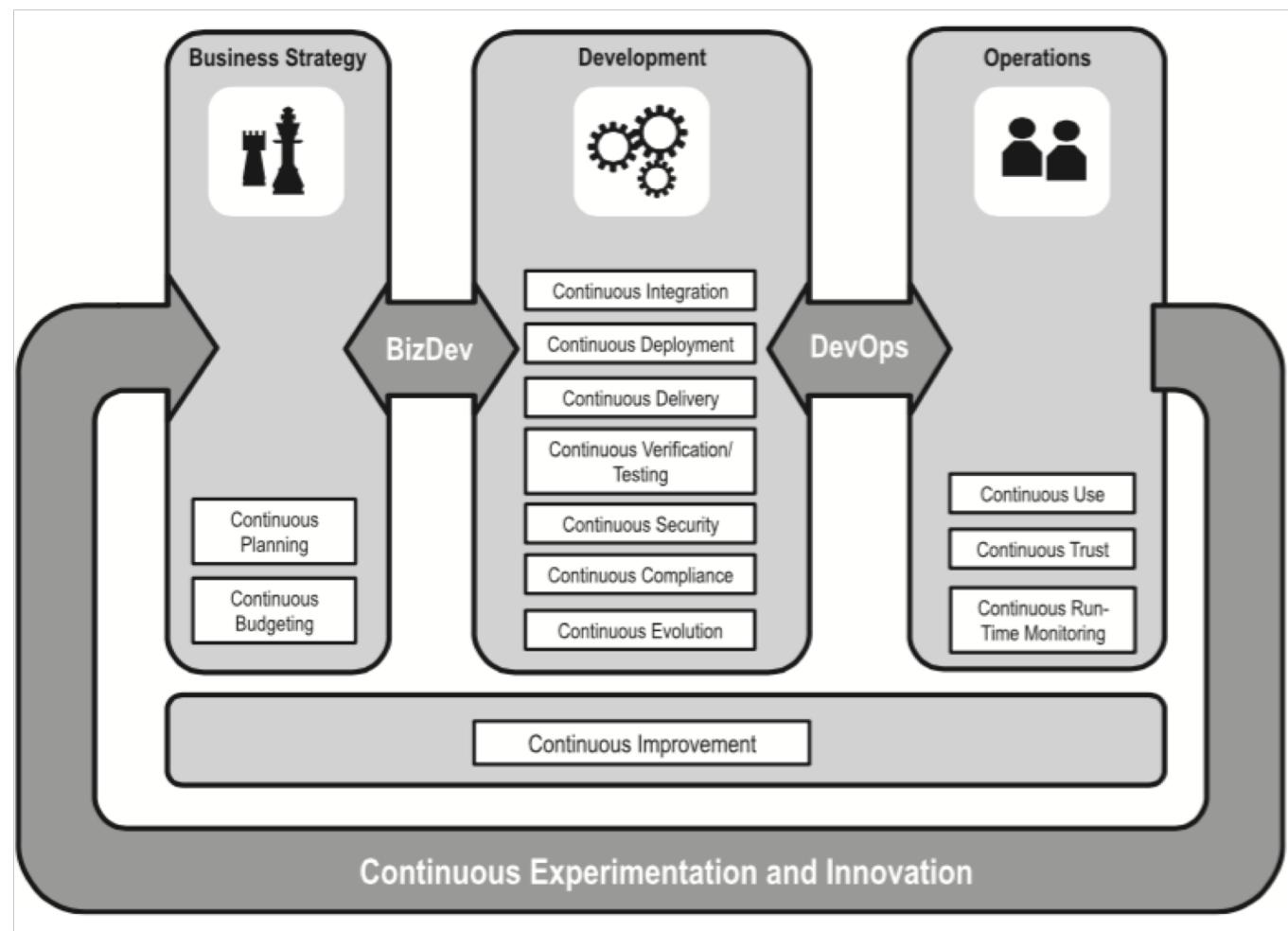
Engenharia de Software Contínua

Ambientes de Desenvolvimento Contínuo

- O desenvolvimento contínuo reconhece que as atividades do processo de desenvolvimento de software devam ser gerenciadas a partir de uma visão mais holística.
- A Base Filosófica, conceitual e técnica: Métodos Ágeis; Desenvolvimento Lean; Lean Startup; DevOps e Software Livre.
- Mentalidade: cadência de fluxo contínuo e não mais como um conjunto de fases, atividades e tarefas, organizadas e executadas de forma discreta

Engenharia de Software Contínua

Ambientes de Desenvolvimento Contínuo



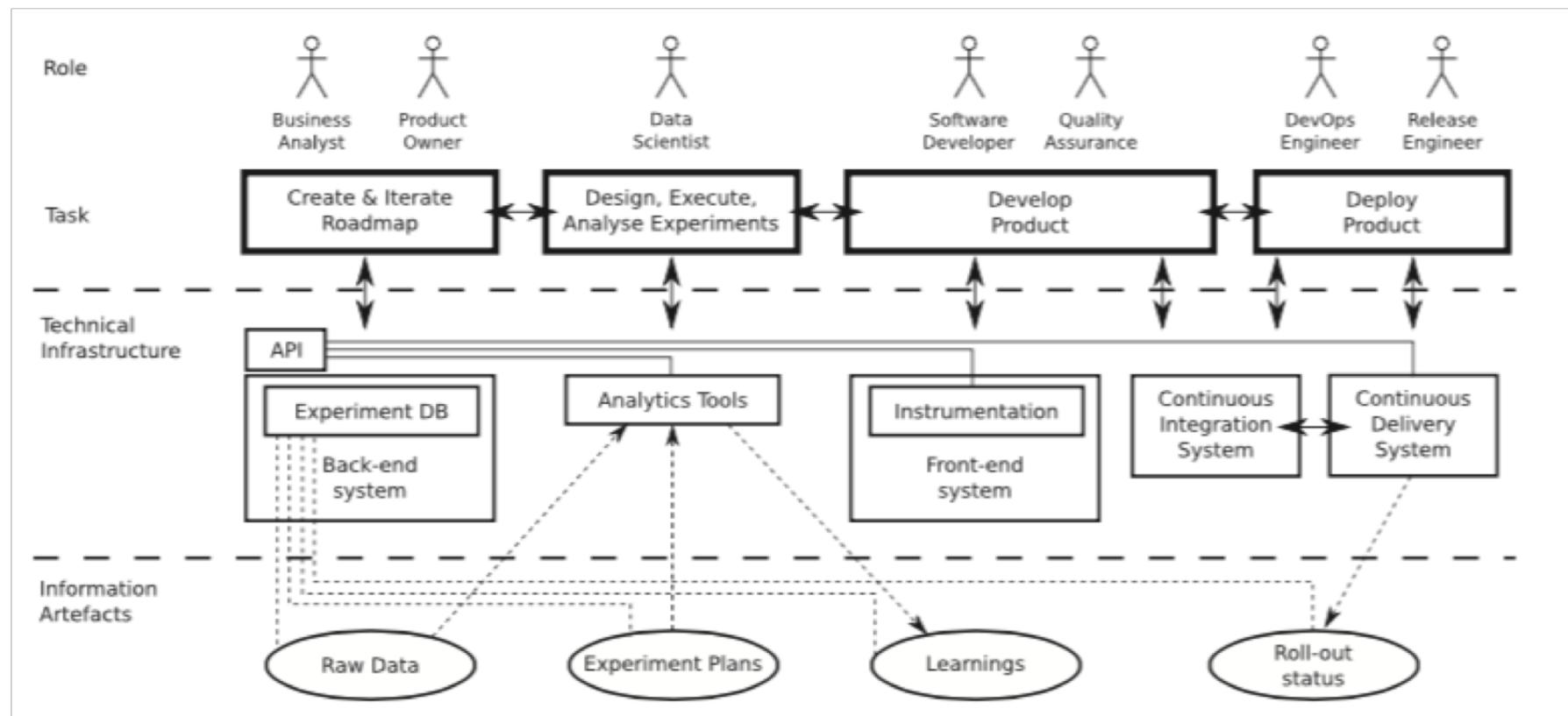
Engenharia de Software Contínua

Ambientes de Experimentação Contínua

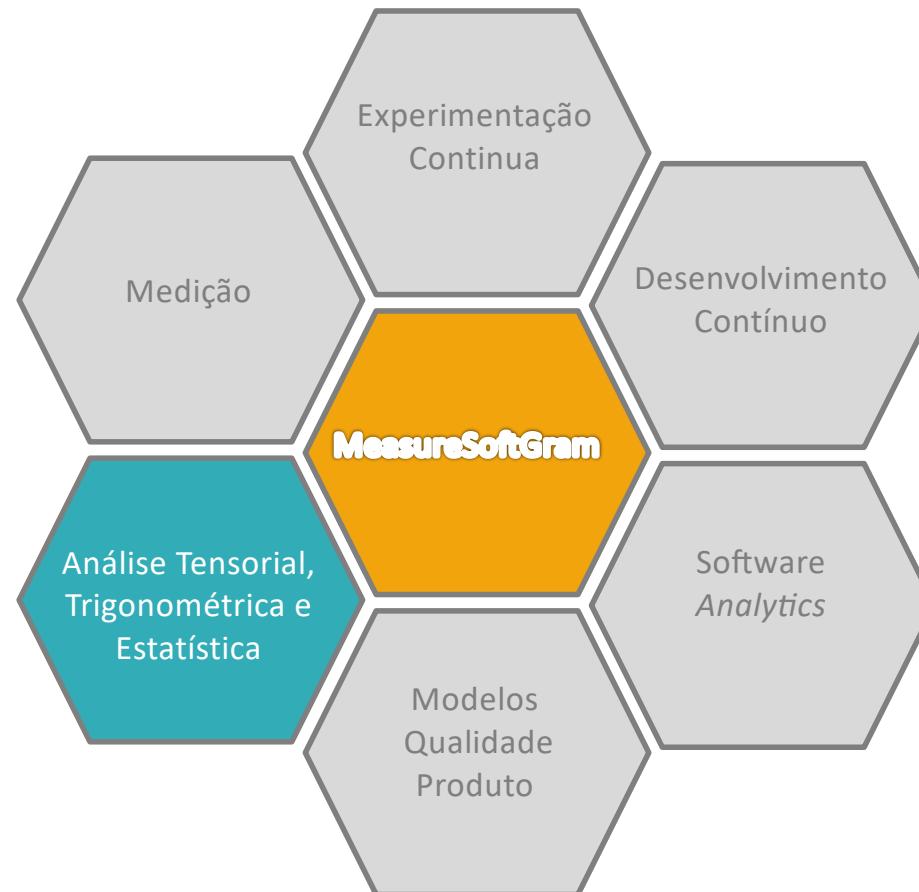
- A idéia básica é desenvolver uma nova versão de um produto(tratamento) e compará-la com a versão em uso(controle), por meio da execução de experimentos controlados.
- Inicialmente apresentada por Kohavi et al. (2009) no contexto do Microsoft Research Group e a Universidade de Stanford:
 - Guia Prático
<https://ai.stanford.edu/~ronnyk/2009controlledExperimentsOnTheWebSurvey.pdf>
- Infraestrutura para experimentação contínua apresentada Diane et al. (2010) no contexto do Google Research.
 - Kubernetes: <https://kubernetes.io/>
- Framework para execução de experimentos on-line apresentado por Bakshy et al. (2014), no contexto do Facebook.
 - Plan-Out: <https://facebook.github.io/planout/>
- Principais práticas apresentadas Schermann et al. (2018) por meio de um survey

Engenharia de Software Contínua

Ambientes de Experimentação Contínua



Revisão da Literatura



Espaço Tensorial

- Generalização dos conceitos de vetores e escalares.
Logo, um tensor é um array multidimensional
 - Dado dois vetores $a \in \mathbb{R}^I$ e $b \in \mathbb{R}^J$, seu produto externo é uma matriz $I \times J$ representado por:
 - $a \circ b$
 - suas entradas (i, j) são dadas por $a(i)b(j)$
 - Essa definição pode ser estendida a um número arbitrário de vetores.
- As Dimensões são conhecidas como modos.
- A dimensionalidade de um tensor é definida pela sua ordem, representada por seus modos.
 - Portanto, um vetor é de um caso especial de um tensor de ordem 0, assim como, um tensor de 3 dimensões possui modo 3.

Espaço Tensorial

- Aplicação na área de computação se mostrou eficiente e já é considerada madura. (Papalexakis, 2016). Ex:
 - visão computacional (Dimitri et al., 2010)
 - processamento de linguagem natural (KaiWei et al., 2013);
 - cidades inteligentes (Wang et al., 2014)
 - análise de dados em redes sociais (Hansen et al., 2015);
 - mineração de dados na web (Agrawal et al., 2015)
- Na área de medicina, como por exemplo, em estudos de sinais cerebrais em análises de exames de eletroencefalograma (Acar et al., 2007).
- Essa gama de aplicação e uso de tensores se deve sua habilidade em representar e modelar dados multimodais ou multiaspectos.

Espaço Tensorial

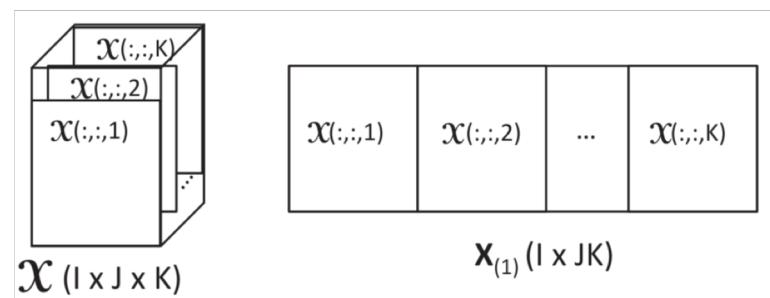
Operações com Tensores

$$\mathbf{A} \otimes \mathbf{B} = \begin{bmatrix} \mathbf{A}(1, 1)\mathbf{B} & \cdots & \mathbf{A}(1, j)\mathbf{B} & \cdots & \mathbf{A}(1, J)\mathbf{B} \\ \vdots & \cdots & \vdots & \cdots & \vdots \\ \mathbf{A}(i, 1)\mathbf{B} & \cdots & \mathbf{A}(i, j)\mathbf{B} & \cdots & \mathbf{A}(i, J)\mathbf{B} \\ \vdots & \cdots & \vdots & \cdots & \vdots \\ \mathbf{A}(I, 1)\mathbf{B} & \cdots & \mathbf{A}(I, j)\mathbf{B} & \cdots & \mathbf{A}(I, J)\mathbf{B} \end{bmatrix}$$

(a) Produto Tensorial (Kronecker)

$$\|\mathcal{X}\|_F = \sqrt{\sum_{i_1}^{I_1} \sum_{i_2}^{I_2} \cdots \sum_{i_N}^{I_N} \mathcal{X}(i_1, i_2, \dots, i_n)^2}$$

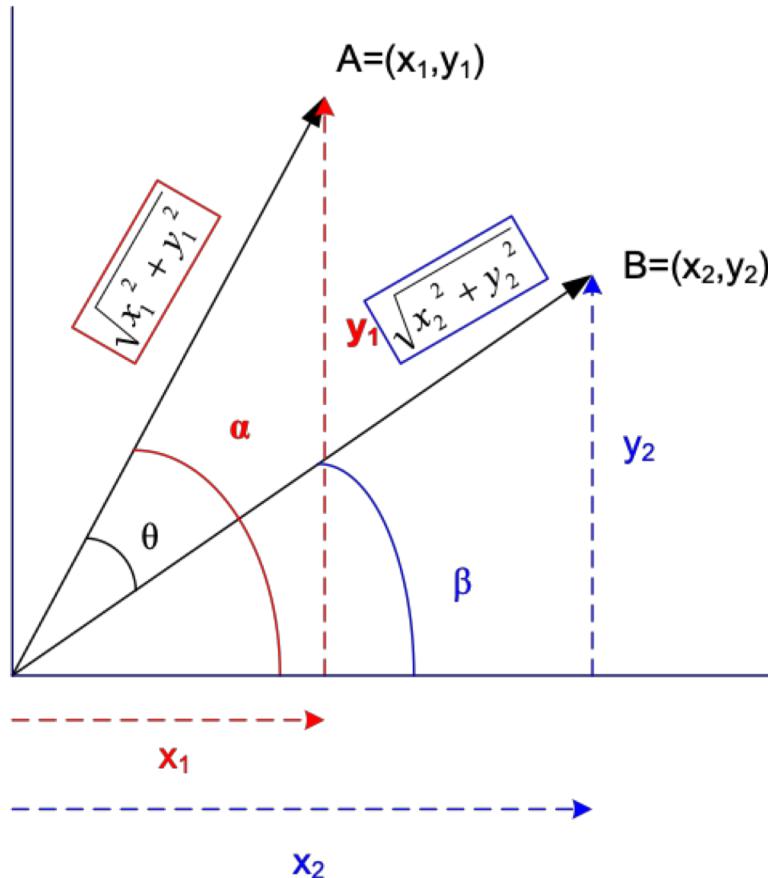
(b) Norma Tensorial (Frobenius)



(c) Vetorização de uma dimensão do Tensor

Similaridade entre Vetores

Diferença Cosseno



$$\cos(\alpha - \beta) = \cos(\alpha)\cos(\beta) + \sin(\alpha)\sin(\beta)$$

Prova : <http://www.themathpage.com/aTrig/sum-proof.htm>

$$\begin{aligned}\cos(\theta) &= \cos(\alpha - \beta) = \\ &= \cos(\alpha)\cos(\beta) + \sin(\alpha)\sin(\beta)\end{aligned}$$

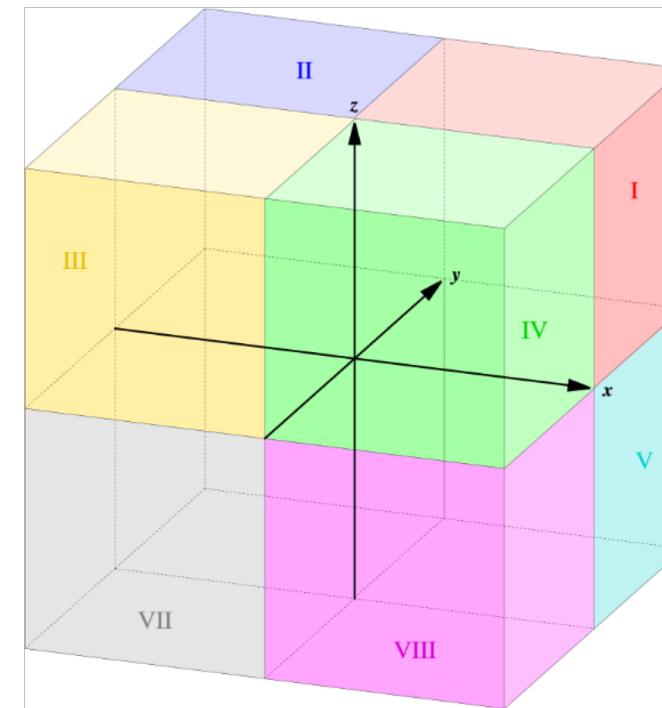
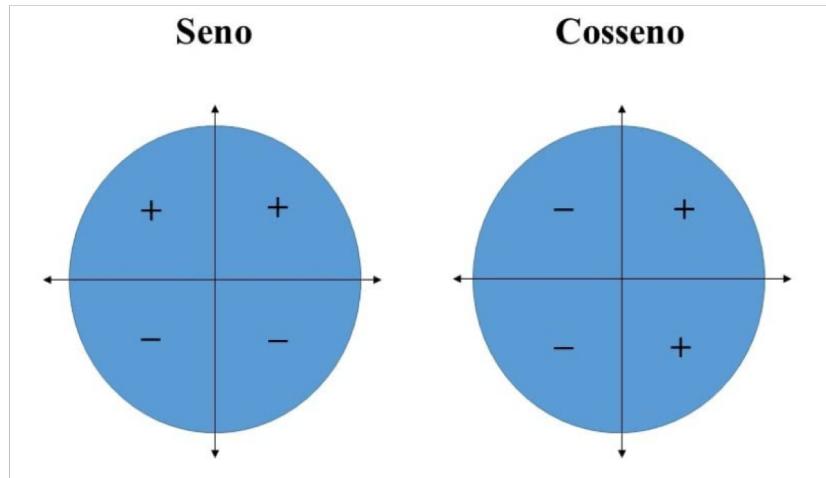
$$\begin{aligned}\cos(\alpha) &= \frac{x_1}{\sqrt{x_1^2 + y_1^2}} & \cos(\beta) &= \frac{x_2}{\sqrt{x_2^2 + y_2^2}}\end{aligned}$$

$$\begin{aligned}\sin(\alpha) &= \frac{y_1}{\sqrt{x_1^2 + y_1^2}} & \sin(\beta) &= \frac{y_2}{\sqrt{x_2^2 + y_2^2}}\end{aligned}$$

$$\begin{aligned}\cos(\theta) &= \\ &= \frac{x_1}{\sqrt{x_1^2 + y_1^2}} \times \frac{x_2}{\sqrt{x_2^2 + y_2^2}} + \frac{y_1}{\sqrt{x_1^2 + y_1^2}} \times \frac{y_2}{\sqrt{x_2^2 + y_2^2}} = \\ &= \frac{x_1 x_2 + y_1 y_2}{\sqrt{x_1^2 + y_1^2} \times \sqrt{x_2^2 + y_2^2}} \\ \cos(\vec{q}, \vec{d}) &= \frac{\vec{q} \bullet \vec{d}}{\|\vec{q}\| \|\vec{d}\|} = \frac{\vec{q}}{\|\vec{q}\|} \bullet \frac{\vec{d}}{\|\vec{d}\|} \\ &\quad \frac{\sum_{i=1}^t x_i y_i}{\sqrt{\sum_{i=1}^t x_i^2 \times \sum_{i=1}^t y_i^2}}\end{aligned}$$

Interpretação da Similaridade

$SPQC_i \text{ (planned)} \times SPQC_i \text{ (realized)}$

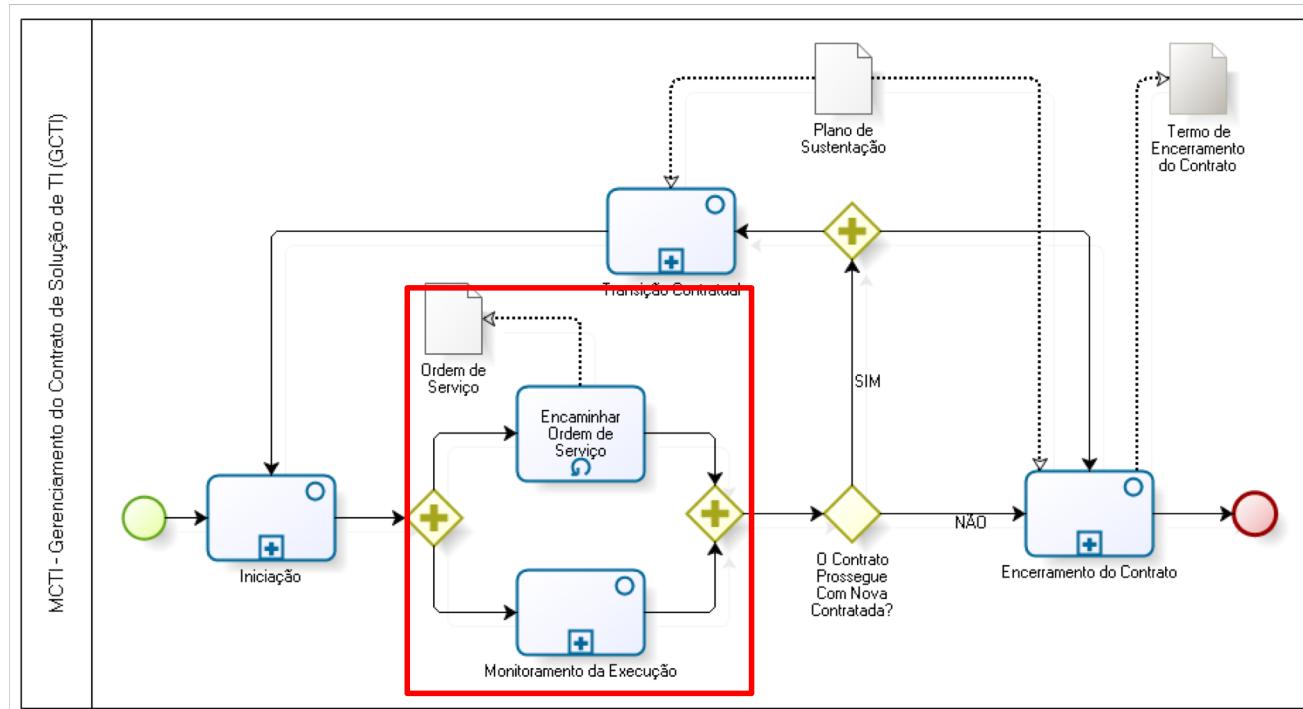


[http://www.wikiwand.com/en/Octant_\(solid_geometry\)](http://www.wikiwand.com/en/Octant_(solid_geometry))

Contratações Públicas Brasileira Federais

Serviços de Desenvolvimento de Sistemas

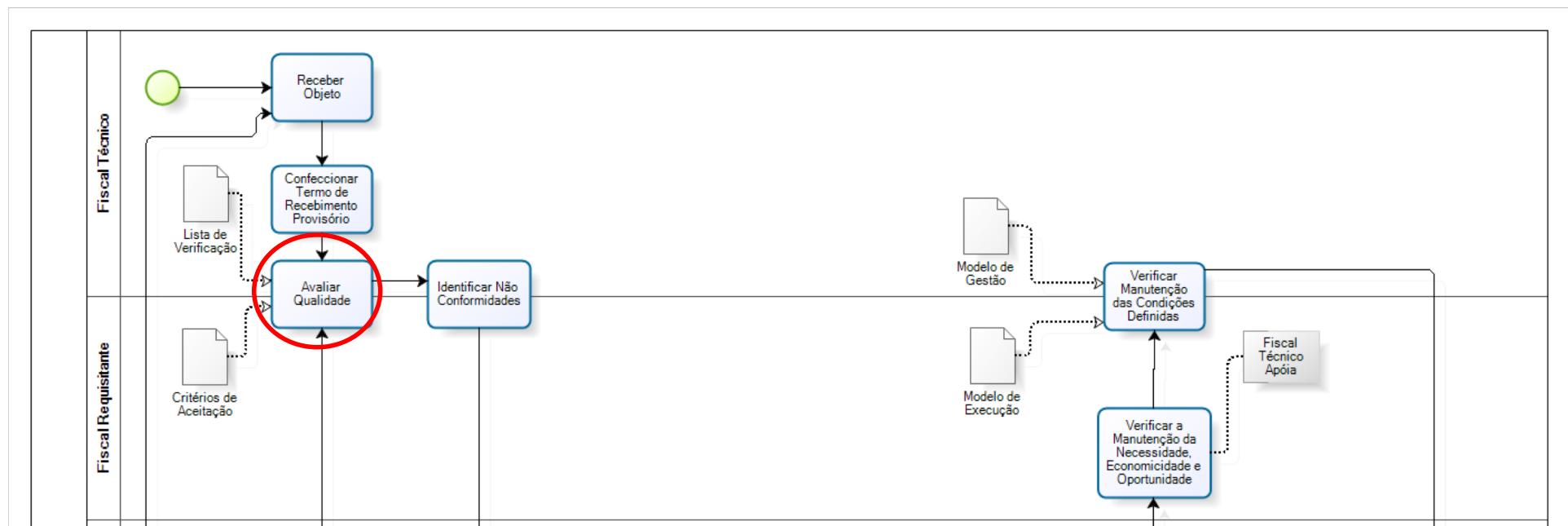
- Lei Nº 8.666, de 21 de Junho de 1993 (Lei8666, 1993)
 - Regulamenta o art. 37, inciso XXI, da Constituição Federal, institui normas para licitações e contratos da Administração Pública e dá outras providências.
- Instrução Normativa Nº 4, de 11 de Setembro de 2014 (IN04, 2010)
 - Dispõe sobre o processo de contratação de Soluções de Tecnologia da Informação pelos órgãos integrantes do Sistema de Administração de Recursos de Tecnologia da Informação e Informática (SISP) do Poder Executivo Federal.



Contratações Públicas Brasileira Federais

Serviços de Desenvolvimento de Sistemas

- Instrução Normativa N° 4, de 11 de Setembro de 2014 (IN04, 2010)
 - Art. 31. A fase de Gestão do Contrato
 - Art. 32. Subseção I - Do início do contrato
 - Art. 33. Subseção II- Do encaminhamento formal de demandas (OS's)
Inciso II - o volume de serviços a serem realizados ou a quantidade de bens a serem fornecidos **segundo as métricas definidas em contrato;**
 - Art. 34. Subseção III - Do monitoramento da execução
Inciso II - **avaliação da qualidade dos serviços** realizados ou dos bens entregues e justificativas, a partir da aplicação das Listas de Verificação e de acordo com os Critérios de Aceitação definidos em contrato, a cargo dos Fiscais Técnico Requisitante do Contrato;

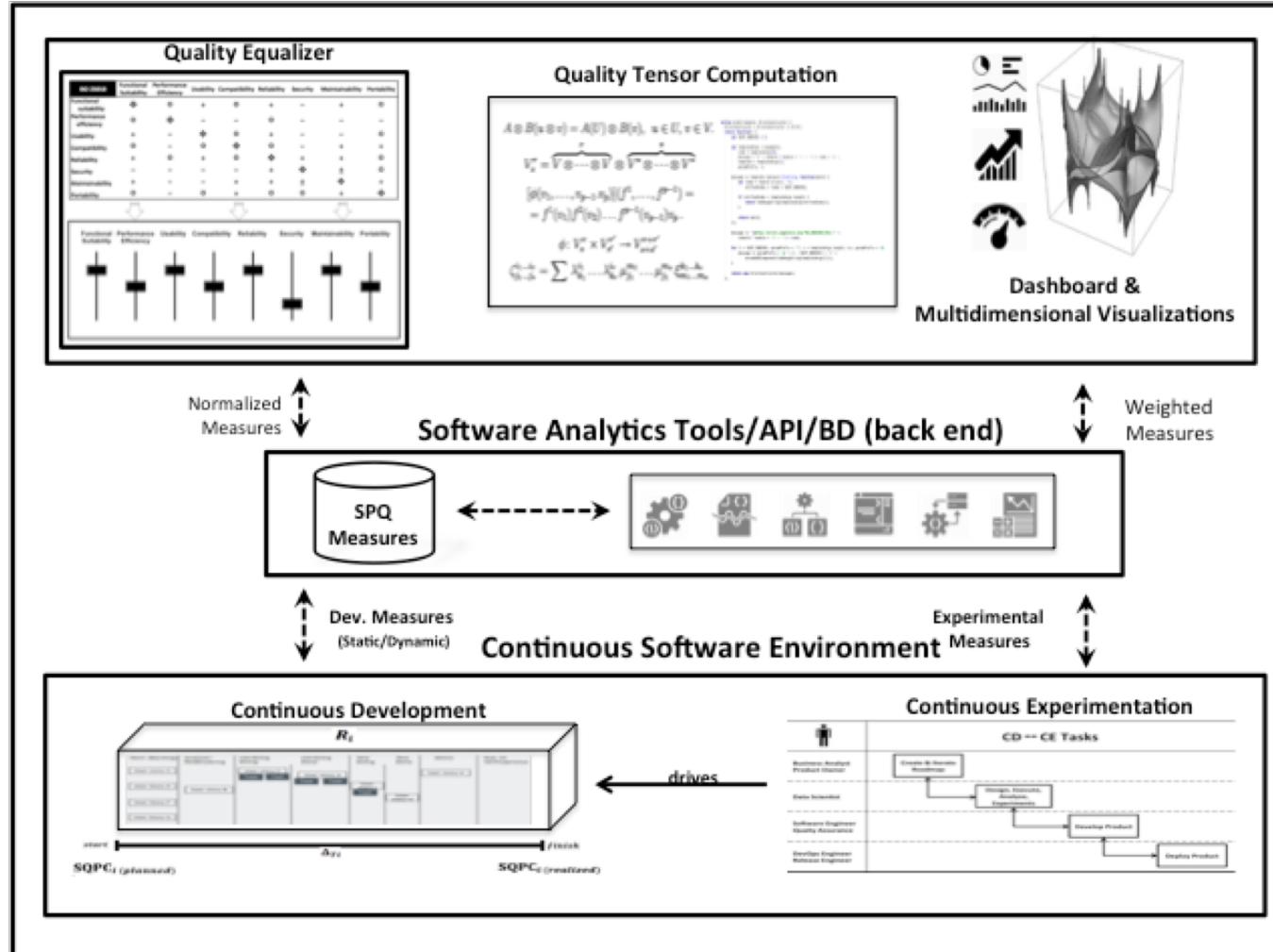


A Proposta

- Considerando:
 - A motivação;
 - A revisão da literatura e trabalhos relacionados;
 - O problema em aberto;
 - O objetivo desta pesquisa e;
 - O contexto.
- Apresentamos a seguir:
 - a proposta
 - o estudo observacional que permitiu explorar a viabilidade da ideia.

A ideia Geral da Proposta

M
e
a
s
u
r
e
s
o
f
t
g



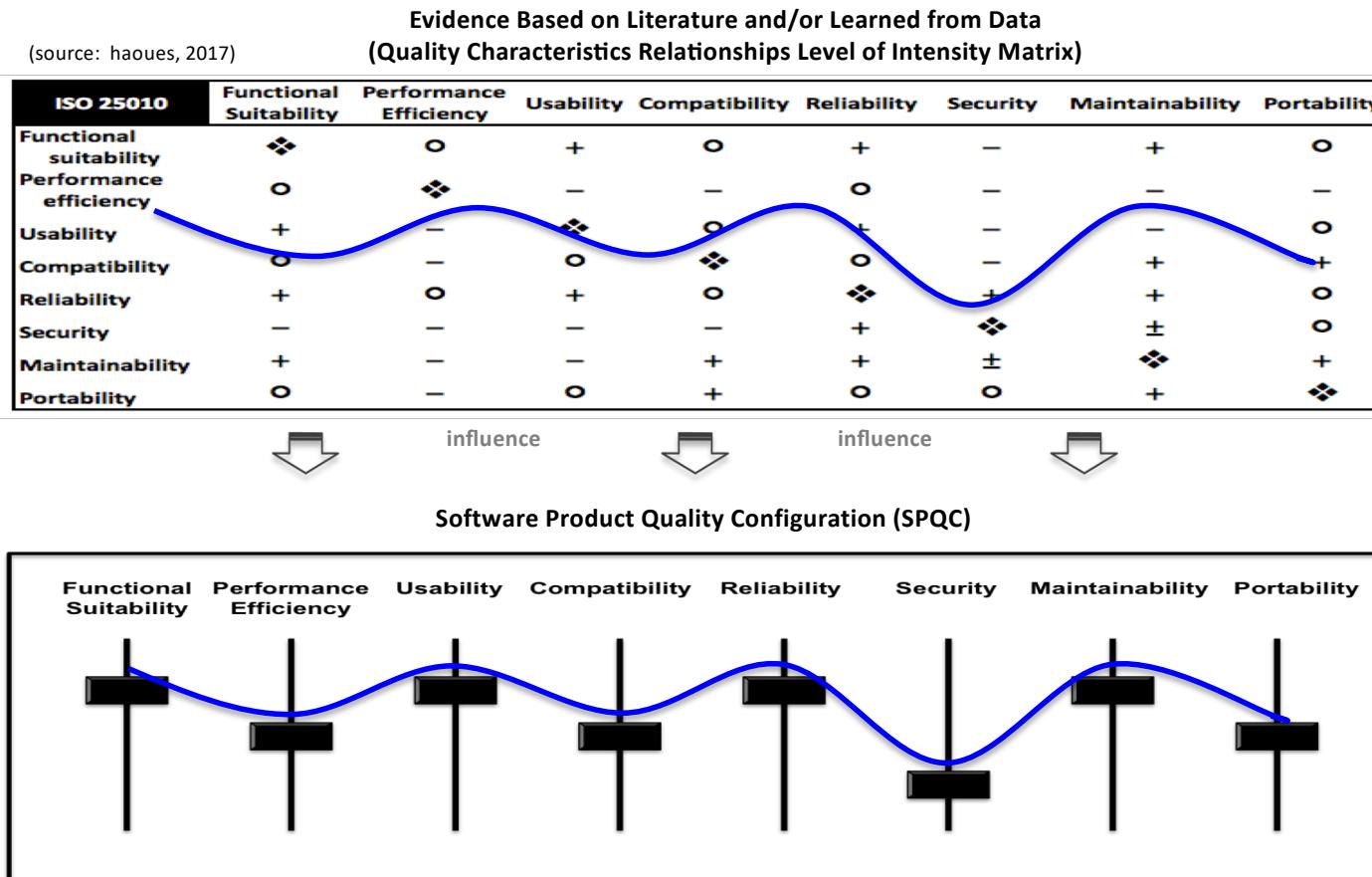
provê suporte
tomada-decisão



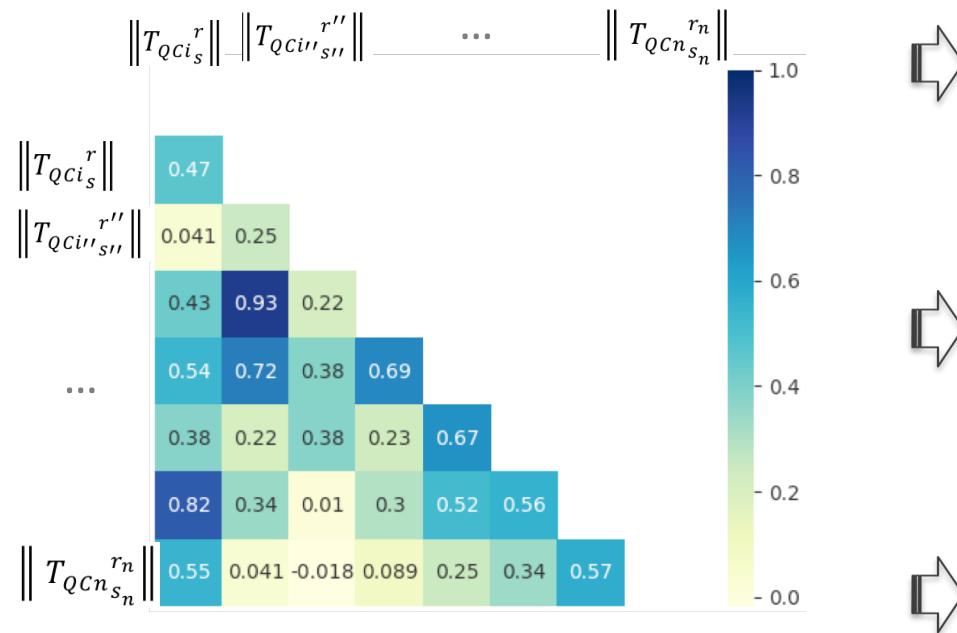
ACEITE do
Release

(para uma versão de produto)

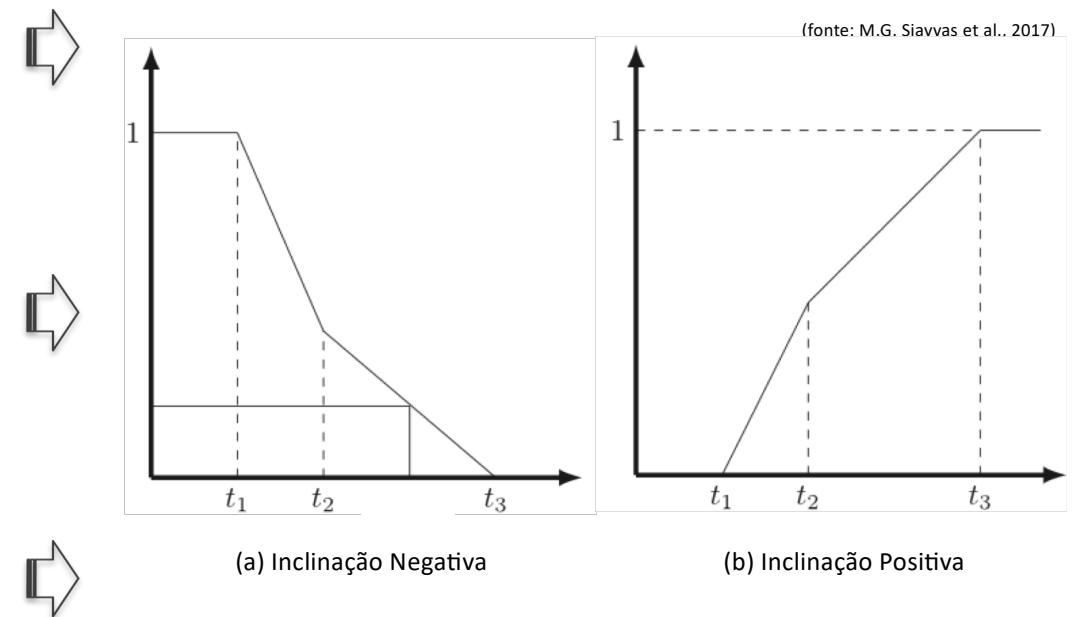
Equalizador de Qualidade



Equalizador de Qualidade

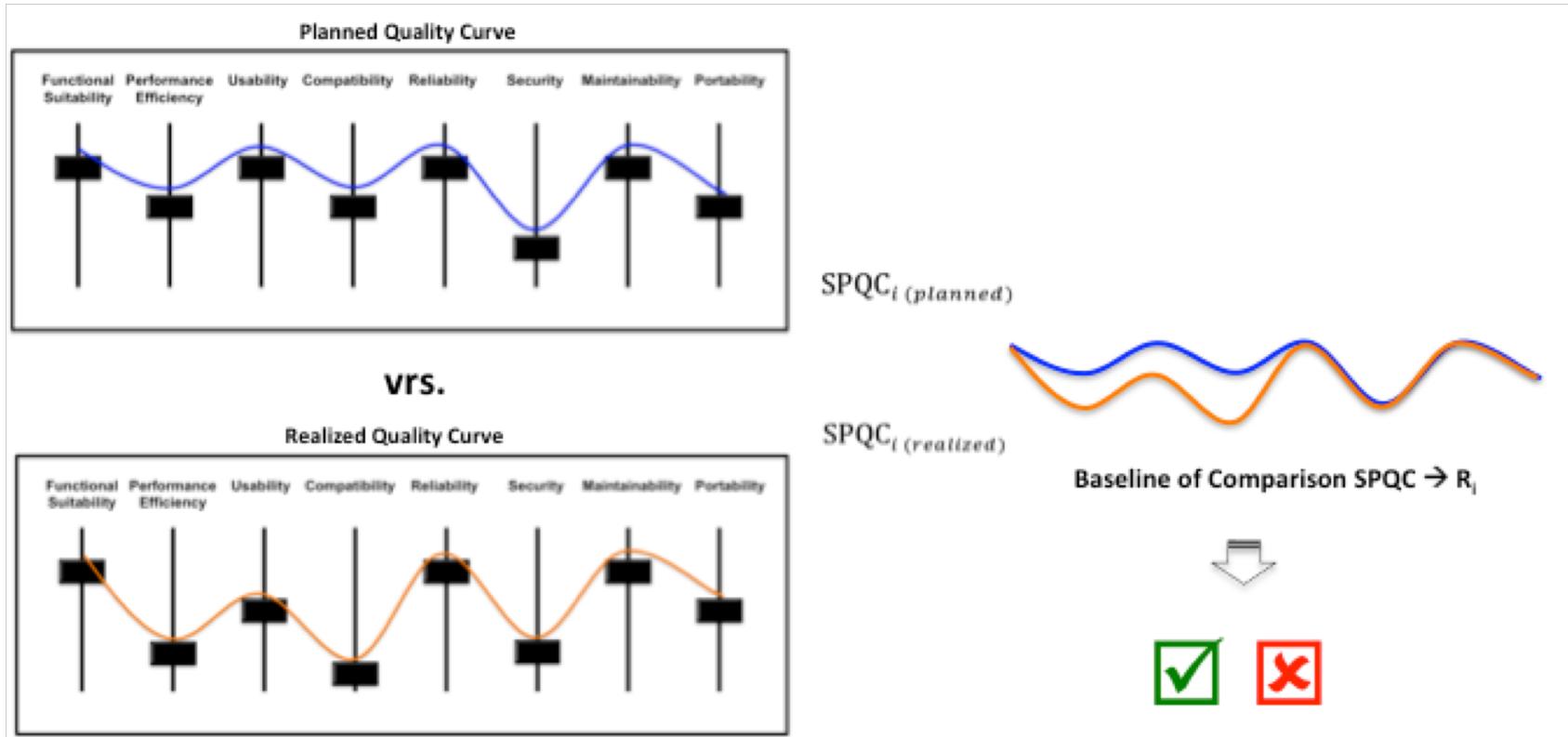


Relações entre as Características da Qualidade reveladas a partir dos dados
(Matriz de Correlação dos tensores de Qualidade)



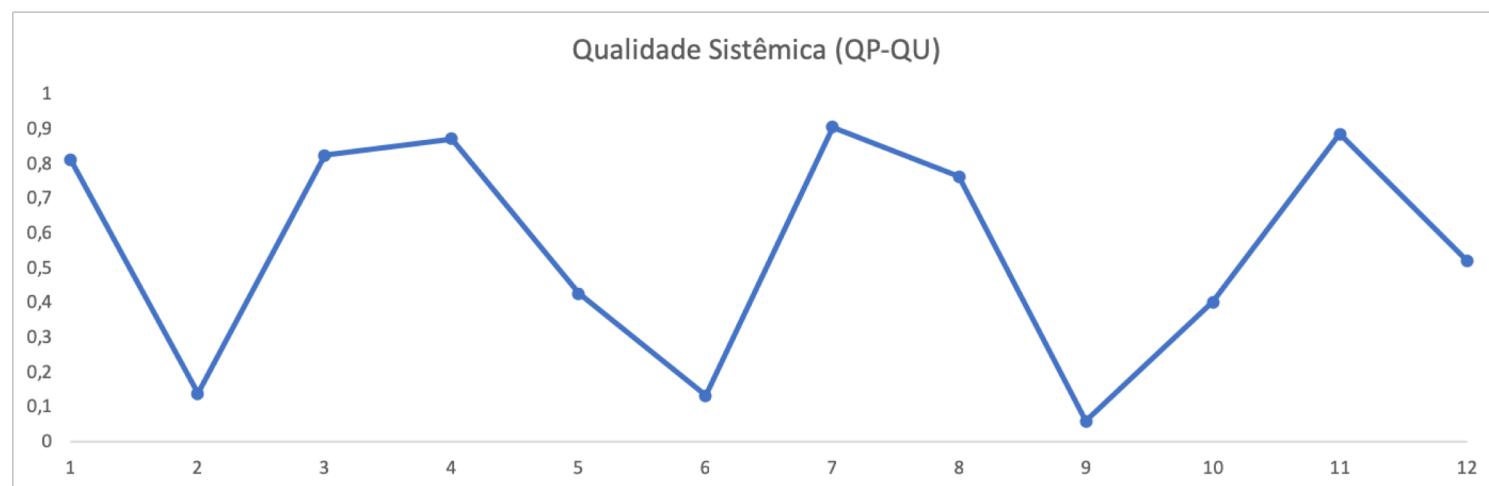
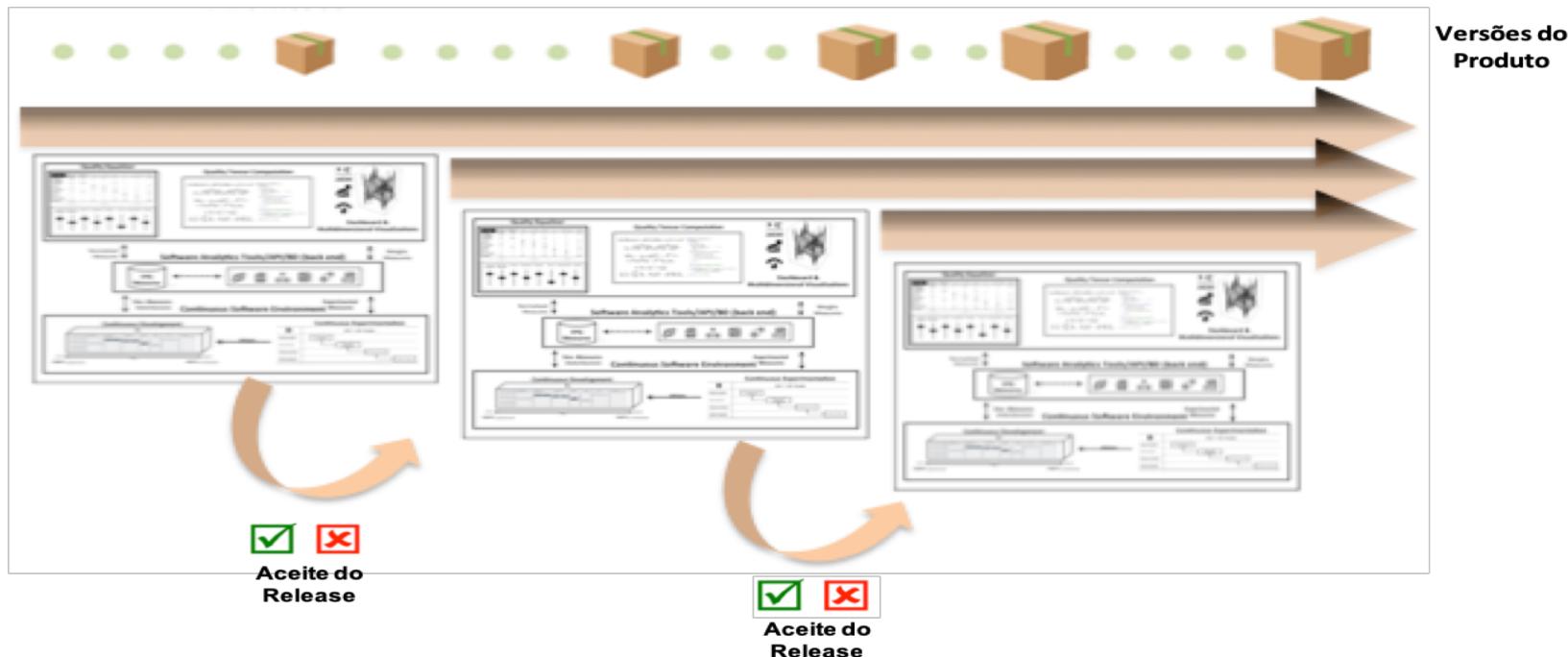
Nível de Intensidade das relações das Características de Qualidade

Equalizador de Qualidade



$$\|SPQC_i\| = \sqrt{\left((T_{QCi_s}^r \otimes W_{QCi}) + (T_{QCi'_s}^{r'} \otimes W_{QCi'}) + (T_{QCi''_s}^{r'} \otimes W_{QCi''}) + \dots + (T_{Qcn_s}^{r_n} \otimes W_{Qcn}) \right)^2}$$

Medição Contínua da Qualidade



Comparação entre as Configurações de Qualidade pela Diferença do Cosseno

Portanto,

$$\cos\left(\overrightarrow{\text{SPQC}_i(\text{planned})}, \overrightarrow{\text{SPQC}_i(\text{realized})}\right) = \frac{\overrightarrow{\text{SPQC}_i(\text{planned})} \bullet \overrightarrow{\text{SPQC}_i(\text{realized})}}{\left|\overrightarrow{\text{SPQC}_i(\text{planned})}\right| \left|\overrightarrow{\text{SPQC}_i(\text{realized})}\right|}$$
$$= \frac{\sum_{i=1}^t x_i y_i}{\sqrt{\sum_{i=1}^t x_i^2 \times \sum_{i=1}^t y_i^2}}$$

Interpretação das Medidas

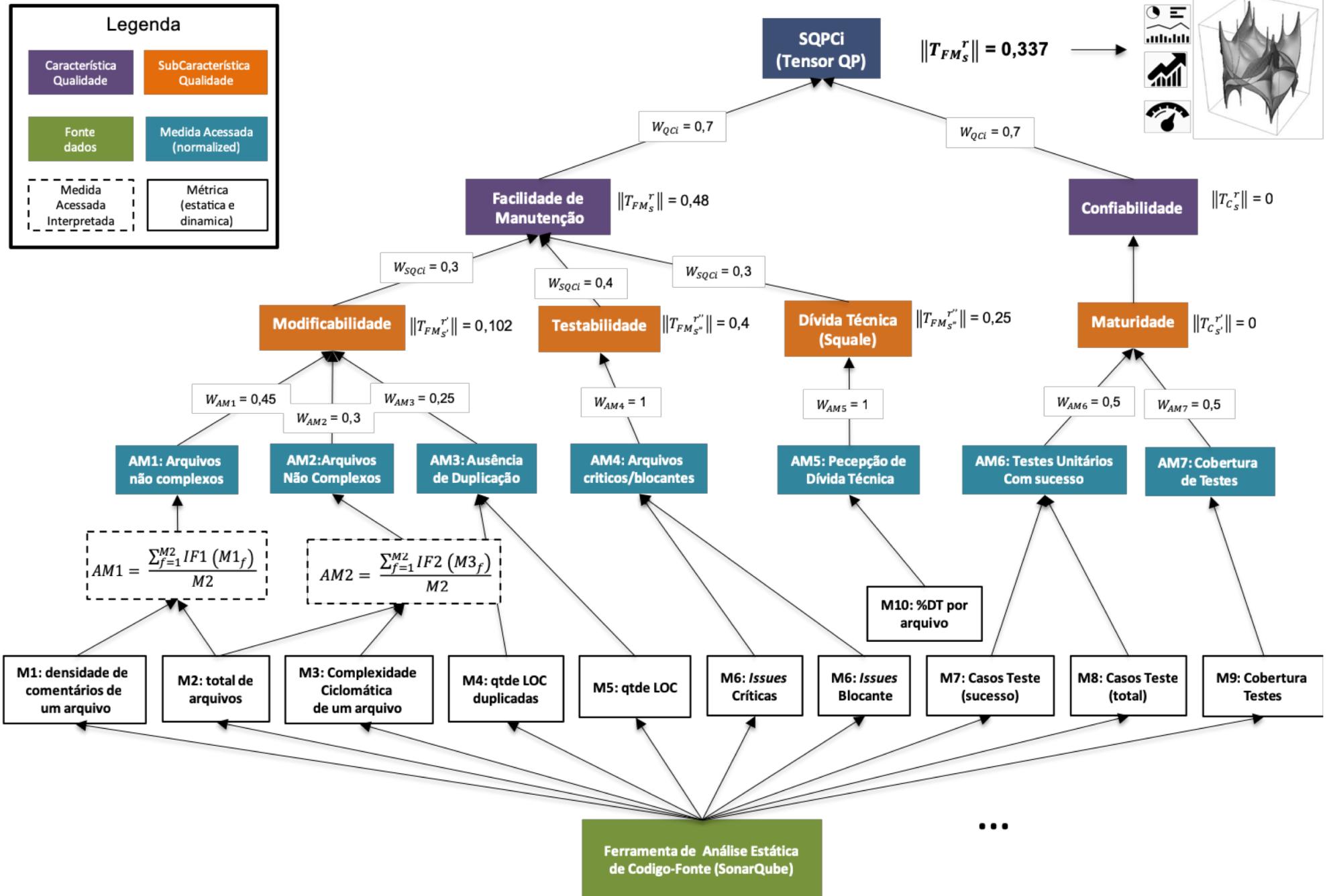
Qualidade de Produto

Config. Qualidade	Característica de Qualidade (Tensor)	Subcarcterística Qualidade	Medida Acessada (normalizada)	Descrição do alerta de qualidade (valores de referência)	Métrica (dado nominal)	Fonte
SPQC	$T_{MTi_s}^r$	Modificabilidade	AM1 = arquivos não complexos	arquivos com a complexidade ciclomática abaixo do threshold=mediana (3,5)	qtde arquivos com CC ≥ 3.5 qtde total de arquivos	SonarQube
			AM2 = comentários em arquivos	arquivos cuja densidade de comentários está fora do limite do threshold= >10% <30%	qtde total LOC qtde LOC comentadas	SonarQube
			AM3 = ausência de duplicação	arquivos acima do threshold $\geq 20\%$ de linhas duplicadas	qtde total LOC qtde LOC duplicadas	SonarQube
		Testabilidade	AM4 = Arquivos críticos/blocantes	issues com severidade classificada como crítica ou blocante. Não é admitta medidas com essa classificação. Valor de referência = 0	qtde de issues críticas qtde de issues blocante	SonarQube
		Dívida Técnica (Squale)	AM5 = Pecepção de Dívida Técnica	arquivos com o percentual de dívida técnica $\geq 30\%$	qtde arquivos com DT ≥ 0.3 qtde total de arquivos	SonarQube
	$T_{C_s}^r$	Maturidade	AM6 = testes unitários com sucesso(<i>passed</i>)	densidade de sucesso na execução dos testes unitários $\geq 95\%$	qtde de casos de teste executados c/ sucesso qtde total de casos de teste	SonarQube
			AM7 = cobertura de teste	código-fonte com cobertura $\geq 40\%$	cobertura de condições e linhas por teste unitário	SonarQube

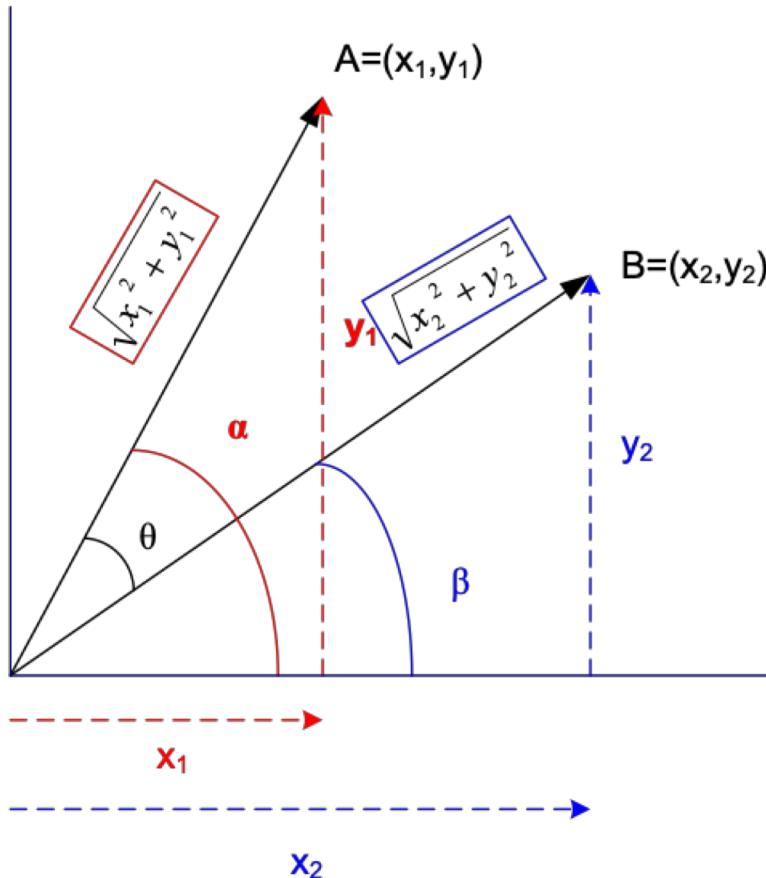
Dados do Parasite (simulados)

Qualidade em Uso

Config. Qualidade	Característica de Qualidade (Tensor)	Subcarcterística Qualidade	Medida Acessada (normalizada)	Descrição do alerta de qualidade (valores de referência)
SPQC	Satisfação	Facilidade de Uso	AM8 = Percepção de Utilidade	Item de questionário usando escala de Lickert (1-5) perguntando ex.: "Como você avalia a utilidade desse sistema no seu contexto de trabalho" [0,1-Muito inútil; 0,25-Inútil; 0,5-Neutro; 0,75-Útil; 1-Muito útil]. Esperado: mediana >= 0,75
		Confiança	AM9 = Potencial de Recomendação	procurar fórmula do Net Promoter Score
		Prazer	AM10 = Percepção de Prazer	Item de questionário usando escala VAS (0-1) perguntando ex.: "O que você achou de usar esse sistema?" [0-Desagradável; 1-Agradável]. Esperado: mediana >= 0,65 (melhor que "neutro")
		Conforto	AM11 = Experiência de Leitura	Item de questionário usando escala VAS (0-1) perguntando ex.: "Como foi a experiência de ler os textos na tela" [0-Difícil; 1-Fácil]. Esperado: Esperado: mediana >= 0,65 (melhor que "neutro")
	Eficácia 	Eficácia	AM12 = Percepção de Eficácia	Caminho direto: participante seguiu as mesmas etapas que o caminho previsto/esperado pelos autores do sistema. Ex: Preenchimento de dados de cartão de crédito ao efetuar uma compra.
	Eficiência	Eficiência	AM13 = Percepção de Eficiência	<p>Sucesso na tarefa: pessoa conseguiu comprar o produto proposto? S/N/Parcial = comprou UM produto, mas não o produto proposto. [COUNT]</p> <p>Quantidade de desvios da tarefa prevista que NÃO ajudam a pessoa a concluir a tarefa. Ex: Quantidade de cliques em outros itens do menu quando a pessoa está procurando a opção de "Configurações".</p>



Comparação entre as Configurações de Qualidade pela Diferença Cosseno



$$\cos(\alpha - \beta) = \cos(\alpha)\cos(\beta) + \sin(\alpha)\sin(\beta)$$

Prova : <http://www.themathpage.com/aTrig/sum-proof.htm>

$$\begin{aligned} \cos(\theta) &= \cos(\alpha - \beta) = \\ &= \cos(\alpha)\cos(\beta) + \sin(\alpha)\sin(\beta) \end{aligned}$$

$$\cos(\alpha) = \frac{x_1}{\sqrt{x_1^2 + y_1^2}} \quad \cos(\beta) = \frac{x_2}{\sqrt{x_2^2 + y_2^2}}$$

$$\sin(\alpha) = \frac{y_1}{\sqrt{x_1^2 + y_1^2}} \quad \sin(\beta) = \frac{y_2}{\sqrt{x_2^2 + y_2^2}}$$

$$\begin{aligned} \cos(\theta) &= \\ &= \frac{x_1}{\sqrt{x_1^2 + y_1^2}} \times \frac{x_2}{\sqrt{x_2^2 + y_2^2}} + \frac{y_1}{\sqrt{x_1^2 + y_1^2}} \times \frac{y_2}{\sqrt{x_2^2 + y_2^2}} = \end{aligned}$$

$$= \frac{x_1 x_2 + y_1 y_2}{\sqrt{x_1^2 + y_1^2} \times \sqrt{x_2^2 + y_2^2}}$$

$$\cos(\vec{q}, \vec{d}) = \frac{\vec{q} \bullet \vec{d}}{\|\vec{q}\| \|\vec{d}\|} = \frac{\vec{q}}{\|\vec{q}\|} \bullet \frac{\vec{d}}{\|\vec{d}\|} = \frac{\sum_{i=1}^t x_i y_i}{\sqrt{\sum_{i=1}^t x_i^2 \times \sum_{i=1}^t y_i^2}}$$

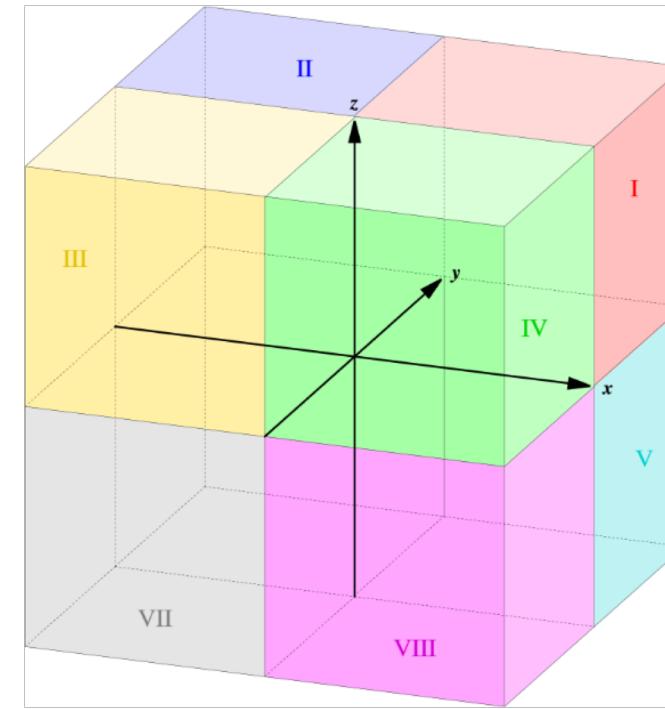
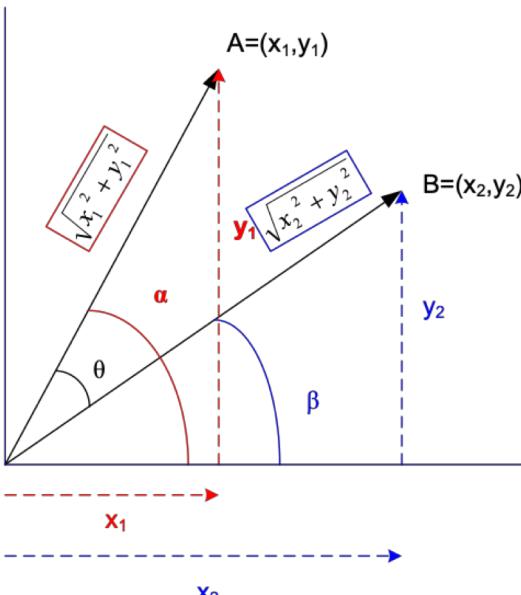
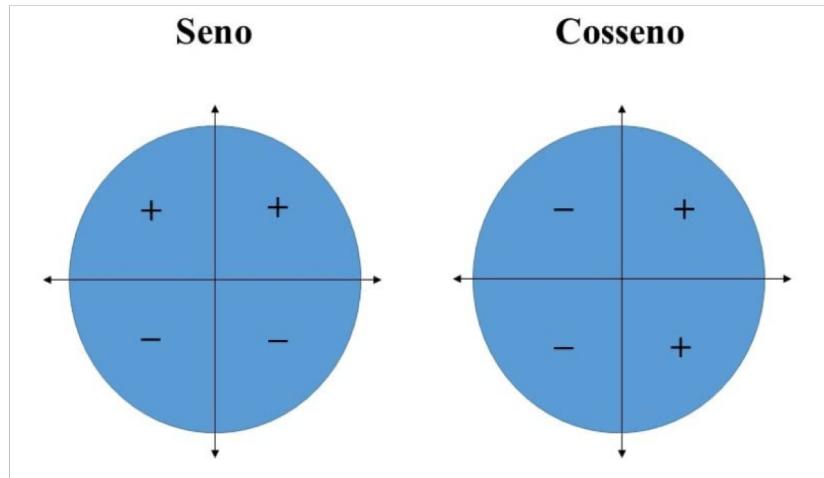
Comparação entre as Configurações de Qualidade pela Diferença do Cosseno

- Portanto,

$$\cos\left(\overrightarrow{\text{SPQC}_i(\text{planned})}, \overrightarrow{\text{SPQC}_i(\text{realized})}\right) = \frac{\overrightarrow{\text{SPQC}_i(\text{planned})} \bullet \overrightarrow{\text{SPQC}_i(\text{realized})}}{\left|\overrightarrow{\text{SPQC}_i(\text{planned})}\right| \left|\overrightarrow{\text{SPQC}_i(\text{realized})}\right|}$$
$$= \frac{\sum_{i=1}^t x_i y_i}{\sqrt{\sum_{i=1}^t x_i^2 \times \sum_{i=1}^t y_i^2}}$$

Interpretação da Comparação

$SPQC_i \text{ (planned)} \times SPQC_i \text{ (realized)}$



[http://www.wikiwand.com/en/Octant_\(solid_geometry\)](http://www.wikiwand.com/en/Octant_(solid_geometry))

Interpretação da Comparação

$\text{SPQC}_i \text{ (planned)} \times \text{SPQC}_i \text{ (realized)}$

- Portanto:
 - Há um indício de que a percepção da qualidade de produto possa ser modelada em tensores comparáveis no espaço
 - A comparação entre tensores da qualidade de produto é feita pela medida de similaridade entre vetores, calculada pela diferença dos cossenos de tais vetores
 - A interpretação do espaço dos vetores no círculo trigonométrico pode revelar tratamentos de anomalias no modelo, como por exemplo, medidas negativas
 - A medida de similaridade simplifica a interpretação

Estudo de Viabilidade

Projeto Parasite

- Sistema de software para apoiar o diagnóstico de doenças tropicais negligenciadas
 - Projeto em parceria do laboratório LENS/COPPE com a Fundação FioCruz
- Desenvolvido por alunos de graduação e pós-graduação da UFRJ
- 1 Versão - 1.081 LOC – 18 arquivos – 28 classes – 9 diretórios – 120 *commits* – Linguagem Python

<https://www.mrdevops-gitlab.com/ParasiteWatch/ParasiteWatch/tree/master>

Estudo de Viabilidade

Projeto Parasite - Instrumentação

- **Instrumentação**

- 1 Máquina virtual (VM) – Linux Debian(stretch) AMD64, 4GB RAM, 2 núcleos de processadores, 1GB armazenamento. Nessa VM estão instaladas as ferramentas SonarQube 7.6, Gitlab 11.19, o repositório do projeto Parasite, além de vários outros pacotes. Essa VM possibilita a execução da etapa de extração das métricas estáticas do repositório do projeto.
- Bibliotecas para manipulação de tensores, além de tratamentos matemáticos e estatísticos que apoiam a análise quantitativa e qualitativa dos dados:
 - TensorLy 0.4.3 - <http://tensorly.org/stable/home.html>
 - Scipy 1.2.1 - <https://www.scipy.org/>
 - NumPy 1.16.2 - <http://www.numpy.org/>
 - Pandas 0.23.4 - <http://pandas.pydata.org/>
 - Python 3.7.0 - <https://www.python.org/downloads/release/python-370/>
 - Basic Linear Algebra Subprograms(BLAS) - <http://www.netlib.orgblas/>
 - Planilhas eletrônicas
 - Arquivos do tipo .cvs (datasets)

Estudo de Viabilidade

Projeto Parasite - Instrumentação

- Critérios de Observação

- Como se trata de uma observação de releases já disponibilizadas, então não há muitas restrições a serem observadas. Destaca-se como impedimento para observação a falta de disponibilidade de todos os ambientes computacionais necessários à instrumentação

- Vantagens e Desvantagens

- A principal vantagem da instrumentação proposta é o apoio semi-automatizado, o que poderia, em última instância, inviabilizar a observação proposta.
 - Já a desvantagem, deve-se ao fato dessa instrumentação possibilitar que apenas o participante do estudo seja analisado.

Estudo de Viabilidade

Projeto Parasite – Painel Geral

SonarQube Projects Issues Rules Quality Profiles Quality Gates

Parasite Watch Web master April 2, 2019, 2:36 AM Version 1.0

Overview Issues Security Reports Measures Code Activity

Quality Gate Passed

Bugs 0 A Vulnerabilities 1 B

Code Smells 42min 6 Debt 0.0% Coverage 0.0% Duplications 0 Duplicated Blocks

About This Project

No tags

1.1K Lines of Code Python 1.1k PHP 21

Project Activity

April 2, 2019 1.0 Show More

Quality Gate (Default) Sonar way

Quality Profiles (PHP) Sonar way (Python) Sonar way

Project Key ParasiteWatch:Web Copy

Get project badges

Embedded database should be used for evaluation purposes only
The embedded database will not scale, it will not support upgrading to newer versions of SonarQube, and there is no support for migrating your data out of it into a different database engine.

SonarQube™ technology is powered by SonarSource SA
Community Edition - Version 7.5 (build 20543) - LGPL v3 - Community - Documentation - Get Support - Plugins - Web API - About

Estudo de Viabilidade

Projeto Parasite – Complexidade Ciclomática

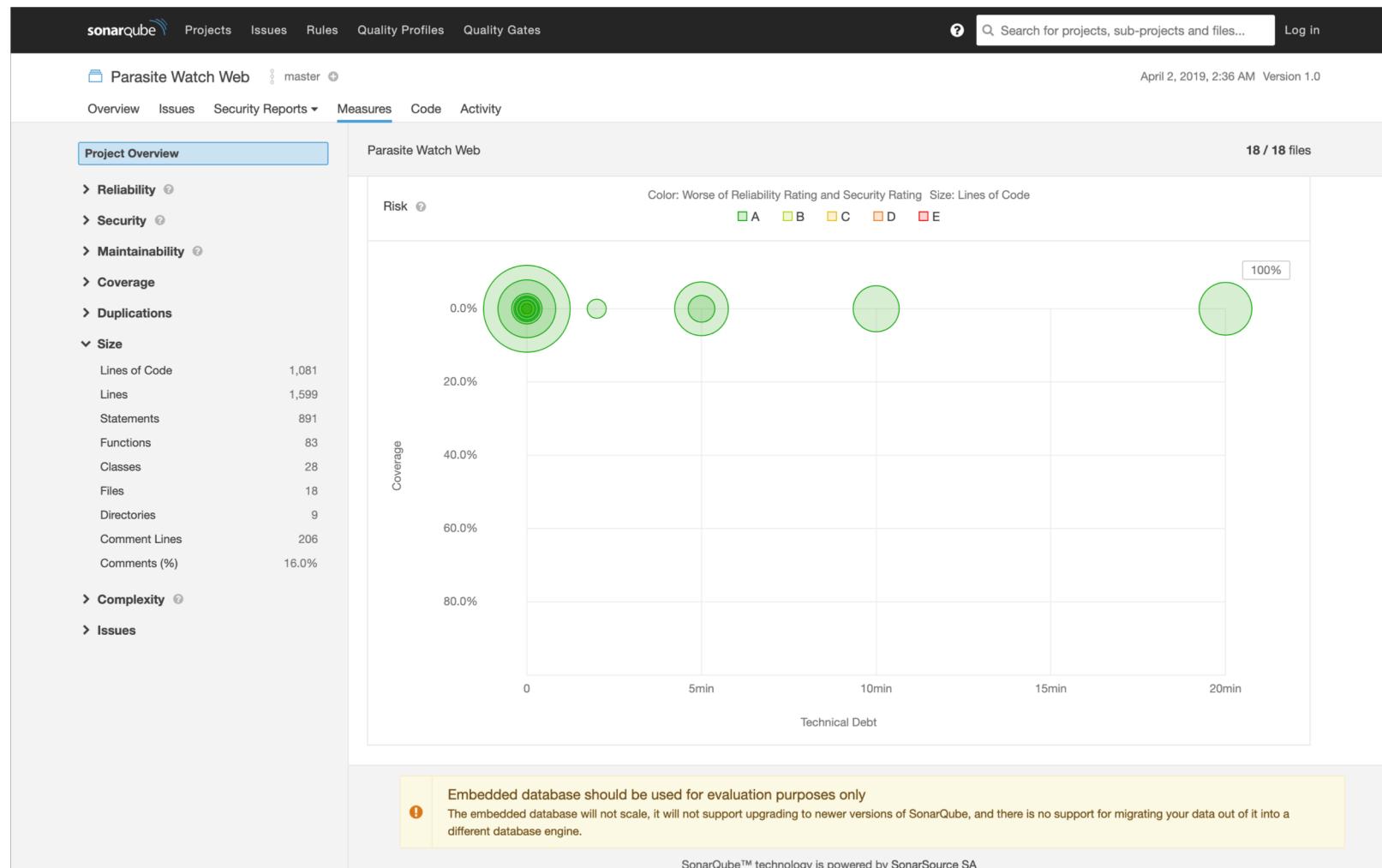
The screenshot shows the SonarQube interface for the 'Parasite Watch Web' project. The top navigation bar includes links for Projects, Issues, Rules, Quality Profiles, and Quality Gates. A search bar and a 'Log in' button are also present. The main content area has tabs for Overview, Issues, Security Reports, Measures (which is selected), Code, and Activity. On the left, a sidebar provides a 'Project Overview' with sections for Reliability, Security, Maintainability, Coverage, Duplications, Size, and Complexity. The Complexity section is expanded, showing 'Cyclomatic Complexity' at 148. Below this, there are sections for Cognitive Complexity (2) and Issues. The right side displays a detailed list of files and their cyclomatic complexity values:

File	Cyclomatic Complexity
app/gathering/views.py	42
test_file.py	25
app/api/views.py	18
app/models.py	17
app/auth/views.py	13
app/gathering/websockets.py	10
app/camera.py	8
migrations/env.py	6
app/__init__.py	5
migrations/versions/37b9a0f4c821_.py	2
app/templates/index/mail.php	1
run.py	1
app/api/__init__.py	0
app/gathering/__init__.py	0
app/auth/__init__.py	0
instance/config.py	0
app/gathering/forms.py	0
app/auth/forms.py	0

At the bottom, a note states: "Embedded database should be used for evaluation purposes only. The embedded database will not scale, it will not support upgrading to newer versions of SonarQube, and there is no support for migrating your data out of it into a different database engine." The footer indicates SonarQube™ technology is powered by SonarSource SA, Community Edition - Version 7.5 (build 20543) - LGPL v3 - Community - Documentation - Get Support - Plugins - Web API - About.

Estudo de Viabilidade

Projeto Parasite – Tamanho



Estudo de Viabilidade

Projeto Parasite – Issues

sonarqube Projects Issues Rules Quality Profiles Quality Gates ? Search for projects, sub-projects and files... Log in

Parasite Watch Web master April 2, 2019, 2:36 AM Version 1.0

Overview Issues Security Reports Measures Code Activity ↑ ↓ to select issues ← → to navigate 1 / 7 issues 1h 12min effort

Filters

Type

- Bug 0
- Vulnerability 1
- Code Smell 6
- Security Hotspot 0

Severity

- Blocker 0
- Critical 0
- Major 1
- Minor 6
- Info 0

Resolution

Status

Creation Date

Language

Rule

Standard

Tag

Module

Directory

File

Assignee

Author

app/__init__.py

Remove the unused local variable "migrate". ... last year L76 9% T- unused

Code Smell Minor Open Not assigned 5min effort

app/api/views.py

Remove the unused local variable "slides". ... 11 months ago L100 9% T- unused

Code Smell Minor Open Not assigned 5min effort

Remove the unused local variable "samples". ... 11 months ago L121 9% T- unused

Code Smell Minor Open Not assigned 5min effort

app/camera.py

Rename this parameter "threadID" to match the regular expression ^[a-z][a-z0-9_]*\$. ... last year L7 9% T- convention

Code Smell Minor Open Not assigned 2min effort

app/models.py

Method "__init__" has 8 parameters, which is greater than the 7 authorized. ... 11 months ago L205 9% T- brain-overload

Code Smell Major Open Not assigned 20min effort

run.py

Make this IP "0.0.0.0" address configurable. ... 11 months ago L9 9% T- cert

Vulnerability Minor Open Not assigned 30min effort

test_file.py

Remove the unused local variable "role". ... last year L43 9% T- unused

Code Smell Minor Open Not assigned 5min effort

7 of 7 shown

Embedded database should be used for evaluation purposes only
The embedded database will not scale, it will not support upgrading to newer versions of SonarQube, and there is no support for migrating your data out of it into a different database engine.

SonarQube™ technology is powered by SonarSource SA
Community Edition - Version 7.5 (build 20543) - LGPL v3 - Community - Documentation - Get Support - Plugins - Web API - About

Dados do Parasite (tabulados)

Qualidade de Produto

Config. Qualidade	Característica de Qualidade (Tensor)	Subcarcteristica Qualidade	Medida Acessada (normalizada)	Cálculo Métrica	Medida Acessada (Normalizada)	Peso Medida	Ponderação Medida (PM)	Peso SubCa. (PSC)	Ponderação Medida-SubCaract.	Agregação SubCaract. (Norma Subespaço Medidas)	Agregação Característica (Norma Subespaço SubCaract.)	Peso Desejado Característica (PEQ)	Ponderação SubCaract. - Caract.	Grau Intensidade Rel. Caract. (haoues)	Agregação SPC (Norma espaço Caract.)
SPQC	$T_{MTI_s}^r$	Manutenibilidade	AM1 = arquivos não complexos	0,500	0,500	0,450	0,225	0,30	0,068	0,102	0,482	70%	0,3374	+	0,337
			AM2 = comentários em arquivos	0,191	0,191	0,300	0,057		0,017						
			AM3 = ausência de duplicação	0	1	0,250	0,250		0,075						
		Testabilidade	AM4 = Arquivos críticos/blocantes	0	1	1,000	1	0,40	0,400	0,4					
		Dívida Técnica (Squale)	AM5 = Percepção de Dívida Técnica	0,167	0,833	1,000	0,833	0,30	0,250	0,25					
	Confiabilidade	$T_{C_s}^r$	AM6 = testes unitários com sucesso(<i>passed</i>)	0	0	0,500	0	1	0	0	0	70%	0	+	
			AM7 = cobertura de teste	0	0	0,500	0		0						

Dados do Parasite (simulados)

Qualidade em Uso

- Cenário de Experimentação Contínua simulado
 - 2 versões de produto de software em uso durante 15 dias, pressupondo aleatorizado do grupo de participantes
 - Duas distribuições não paramétricas geradas de forma aleatória
 - Qualidade em uso percebida segundo mesmo modelo/*rationale* utilizado para perceber a qualidade de produto

Config. Qualidade	Característica de Qualidade (Tensor)	Subcarcterística Qualidade	Medida Acessada (normalizada)	Descrição do alerta de qualidade (valores de referência)
SPQC	Satisfação	Facilidade de Uso	AM8 = Percepção de Utilidade	Item de questionário usando escala de Lickert (1-5) perguntando ex.: "Como você avalia a utilidade desse sistema no seu contexto de trabalho" [0,1-Muito inútil; 0,25-inútil; 0,5-Neutro; 0,75-Útil; 1-Muito útil]. Esperado: mediana >= 0,75
		Confiança	AM9 = Potencial de Recomendação	procurar fórmula do Net Promoter Score
		Prazer	AM10 = Percepção de Prazer	Item de questionário usando escala VAS (0-1) perguntando ex.: "O que você achou de usar esse sistema?" [0-Desagradável; 1-Agradável]. Esperado: mediana >= 0,65 (melhor que "neutro")
		Conforto	AM11 = Experiência de Leitura	Item de questionário usando escala VAS (0-1) perguntando ex.: "Como foi a experiência de ler os textos na tela" [0-Difícil; 1-Fácil]. Esperado: Esperado: mediana >= 0,65 (melhor que "neutro")
	Eficácia	Eficácia	AM12 = Percepção de Eficácia	Caminho direto: participante seguiu as mesmas etapas que o caminho previsto/esperado pelos autores do sistema. Ex: Preenchimento de dados de cartão de crédito ao efetuar uma compra.
		Eficiência	AM13 = Percepção de Eficiência	Sucesso na tarefa: pessoa conseguiu comprar o produto proposto? S/N/Parcial = comprou UM produto, mas não o produto proposto. [COUNT] Quantidade de desvios da tarefa prevista que NÃO ajudam a pessoa a concluir a tarefa. Ex: Quantidade de cliques em outros itens do menu quando a pessoa está procurando a opção de "Configurações".

Dados do Parasite (simulados)

Qualidade em Uso

- Conjunto de Dados(*data sets*)

Satisfação	Eficiência	Eficácia
0.3996707	0.00658941	0.81143045
0.42788762	0.32737993	0.13830564
0.1389098	0.0207637	0.82493763
0.49006457	0.27177696	0.87239129
0.75683177	0.01758895	0.42689382
0.11007855	0.50070249	0.13269444
0.30359668	0.8131612	0.90542066
0.22601737	0.39141474	0.76397512
0.92164487	0.55896791	0.05851286
0.62233116	0.48275795	0.40263632
0.28500867	0.29391252	0.88558596
0.19377024	0.05990281	0.5215296
0.32333908	0.64482978	0.33344411
0.10462794	0.5411605	0.17363779
0.59023642	0.21044627	0.89375521

Versão de produto A - $SPQC_i(A)$

Satisfação	Eficiência	Eficácia
0.8105976	0.85589092	0.03576582
0.36211437	0.27438837	0.89456372
0.70355816	0.34293988	0.16593994
0.16542732	0.34919783	0.33581962
0.10564745	0.59641318	0.31175144
0.12604926	0.53320242	0.3630265
0.29555203	0.45932322	0.80047663
0.81252994	0.99133466	0.56543731
0.52845671	0.00274382	0.86125031
0.04984305	0.84013123	0.55455449
0.53852653	0.72682096	0.46190324
0.26680332	0.00282056	0.59421481
0.04892979	0.8608729	0.31431647
0.25271812	0.56513284	0.76508887
0.3218888	0.62853529	0.94219539

Versão de produto B - $SPQC_i(B)$

Dados do Parasite (simulados)

Qualidade em Uso

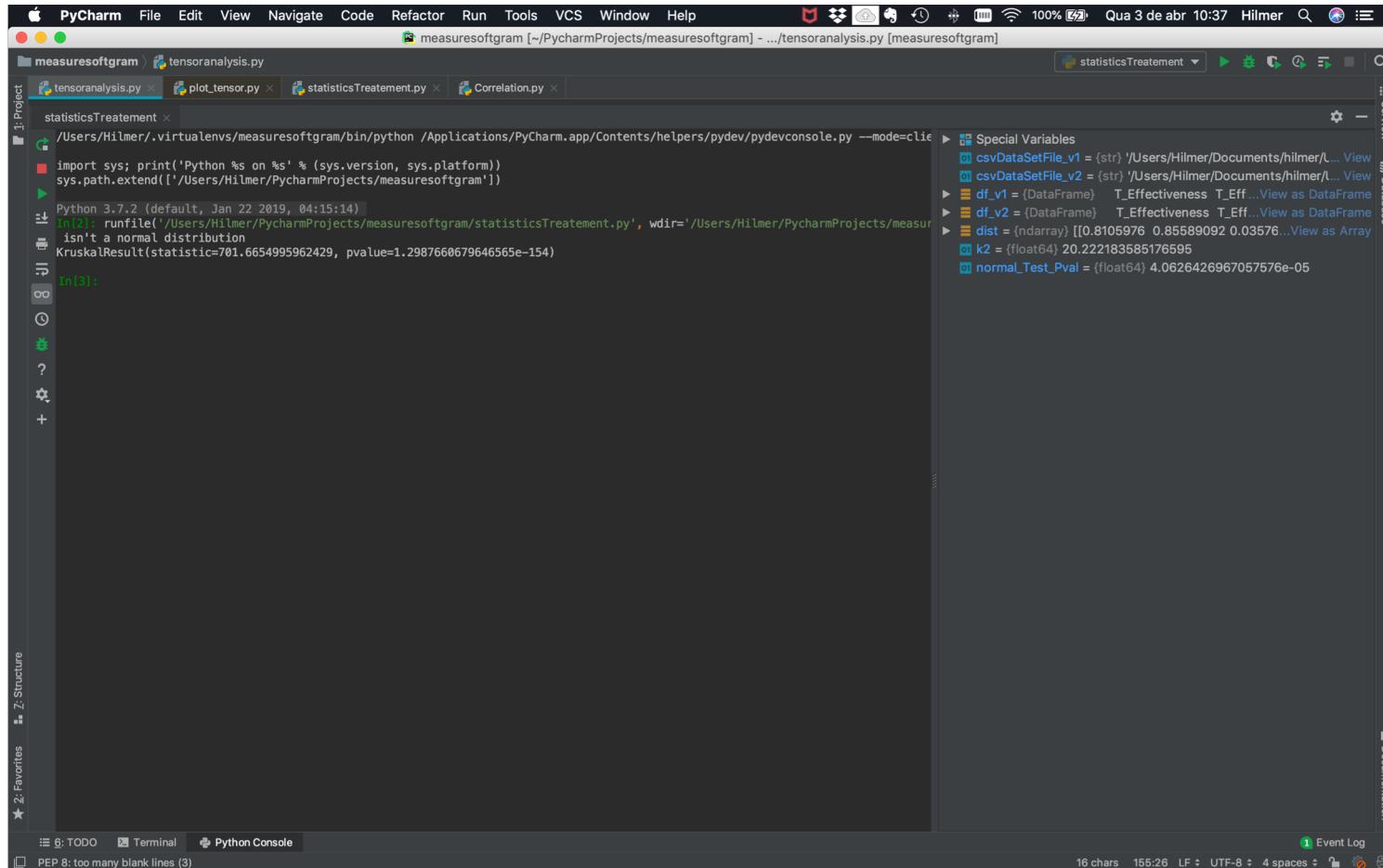
- *Design do Experimento*
 - 1 fator X 2 tratamentos
 - Fator: Qualidade Sistêmica percebida pelo surrogate da configuração de qualidade $SPQC_i$
 - Tratamentos:
 - Versão de Produto A $SPQC_i(A)$
 - Versão de Produto B $SPQC_i(B)$
- Formulação das Hipóteses

$$H_0 : SPQC_i(A) = SPQC_i(B)$$

$$H_1 : SPQC_i(A) > SPQC_i(B)$$

Dados do Parasite (simulados)

Qualidade em Uso



The screenshot shows a PyCharm interface with the following details:

- Terminal:** Displays a Python session output:

```
/Users/Hilmer/.virtualenvs/measuresoftgram/bin/python /Applications/PyCharm.app/Contents/helpers/pydev/pydevconsole.py --mode=cli
import sys; print('Python %s on %s' % (sys.version, sys.platform))
sys.path.extend(['/Users/Hilmer/PycharmProjects/measuresoftgram'])

Python 3.7.2 (default, Jan 22 2019, 04:15:14)
[cl]: runfile('/Users/Hilmer/PycharmProjects/measuresoftgram/statisticsTreatment.py', wdir='/Users/Hilmer/PycharmProjects/measuresoftgram')
isn't a normal distribution
KruskalResult(statistic=701.6654995962429, pvalue=1.2987660679646565e-154)
```
- Variable Explorer:** Shows variables from the current session:
 - Special Variables
 - csvDataSetFile_v1 = (str) '/Users/Hilmer/Documents/hilmer/L... View
 - csvDataSetFile_v2 = (str) '/Users/Hilmer/Documents/hilmer/L... View
 - df_v1 = (DataFrame) T_Effectiveness T_Eff...View as DataFrame
 - df_v2 = (DataFrame) T_Effectiveness T_Eff...View as DataFrame
 - dist = (ndarray) [0.8105976 0.85589092 0.03576...View as Array
 - k2 = {float64} 20.222183585176595
 - normal_Test_Pval = {float64} 4.0626426967057576e-05
- Status Bar:** Shows 16 chars, 155:26, LF, UTF-8, 4 spaces.

KruskalResult(statistic=701.6654995962429, pvalue=1.2987660679646565e-154)

Ameaças a Validade da Pesquisa

- *Constructo*

- A validade de constructo está associada a capacidade da teoria descrita em representar o fenômeno observado. Ainda não podemos fazer nenhuma afirmação a respeito.
- Por outro lado, no intuito de mitigar essa ameaça, desenvolvemos um protótipo funcional utilizando dados da qualidade de produto extraídos de um projeto real, em desenvolvimento.
- Outras ameaças a validade de construção, segundo Wholin, Runeson, et al (2012), são mitigadas neste estudo:
 - *mono method bias* - diferentes tipos de métricas são utilizadas neste estudo
 - *mono operation bias* – várias classes e métodos são analisados
- Precisamos observar mais projetos, além de estudos experimentais junto a indústria!
- A reproduzibilidade desse estudo é totalmente garantida
 - Os dados, o código-fonte do protótipo, além das planilhas eletrônicas estão disponíveis em: <https://gitlab.com/hneri/measuresoftgram/> (será migrado para LENS-MrDevops)
 - Os procedimentos para reprodução da VM utilizada estão disponíveis em: <https://www.mrdevops-gitlab.com/plataforma-de-desenvolvimento-continuo-para-iot/chef>

Ameaças a Validade da Pesquisa

- Interna

- Como o participante deste estudo é um produto de software, onde “versões” já produzidas serão analisadas, muitas das ameaças a validade descritas por Wholin, Runeson, et al (2012), ainda que em ambiente simulado, são observadas como estratégia de mitigação, como por exemplo ameaças sociais e ameaças de múltiplos grupos.
- Por outro lado, deve-se ter especial atenção a algumas ameaças que dizem respeito ao uso de diferentes soluções computacionais. Essa questão está relacionada a ameaça de um grupo simples, principalmente a ambiguidade da influência na direção da causa. Nesse sentido, algumas dessas ameaçadas identificadas estão sendo mitigadas:
 - A estrutura de dados representando tensores (arrays multidimensionais) são os mesmos para representar a qualidade de produto, a qualidade em uso e as respectivas configurações de qualidade
 - Fatores de confusão relacionados a comparação e interpretação das configurações de qualidade procuram ser mitigados por meio de conceitos e aplicações algébricas e trigonométricas
 - A biblioteca TensorLy fornece um conjunto de testes automatizados que podem ser reproduzidos e auditados. <https://github.com/tensorly/tensorly/tree/master/tensorly/tests>
 - As bibliotecas de tratamento algébrico e estatístico, disponibilizadas na linguagem Python têm sido utilizadas em diferentes pesquisas, inclusive em áreas de conhecimento diferentes da ciência da computação e engenharia de software. <http://www.bookmetrix.com/detail/book/c96d4bed-b58c-4fc0-9397-d8254be19de8#citations>
- Por fim, todas as soluções computacionais utilizadas neste estudo es-tão no domínio de software livre. Portanto, como em todo projeto de software livre o código-fonte é aberto, todas as soluções utilizadas neste estudo são passíveis de auditoria a qualquer tempo.

Ameaças a Validade da Pesquisa

- Externa
 - A validade externa está relacionada a capacidade de generalização. Por se tratar de um estudo observacional de caracterização há claramente a presença da ameaça de interação entre os tratamentos e o pesquisador, uma vez que, a amostra do estudo não é representativa frente a população de produtos de software. Principalmente em função de uso parcial de dados simulados
 - Por outro lado, a ameaça de interação entre tratamento e a configuração do estudo é garantida uma vez que o participante é um produto de software que está sendo desenvolvido e vem passando por sucessivas evoluções de melhoria.
 - Além disso, por se tratar da análise de “versões” já produzidas do produto, a ameaça de interação entre o histórico e tratamento é mitigada

Ameaças a Validade da Pesquisa

- Conclusão
 - A ameaça de conclusão está associada a robustez dos testes estatísticos realizados no estudo. Para mitigar esse tipo de ameaça foram definidas:
 - um modelo matemático com aplicações algébricas e trigonométricas que indicam indícios de propriedade canônica. O conjunto de regras de transformação é aplicado em qualquer granularidade do modelo.
 - Implementação de um protótipo funcional com os testes estatísticos propostos no modelo
 - testes de hipóteses do tipo A/B para simular versões de produto em uso, em ambiente experimentação contínua.

Conclusões

Macroatividades Realizadas

- Pesquisa bibliográfica sobre as áreas de conhecimento que suportam a ideia desta proposta. Quer sejam:
 - a) modelos qualidade de produto de software;
 - b) visão analítica para observar e apoiar a tomada de decisão sobre a qualidade de produto de software;
 - c) ambientes de desenvolvimento e experimentação contínua de software para apoiar a tomada de decisão sobre a escolha de versões de produtos de produtos de software;
 - d) uso de tensores, aplicações algébricas e trigonométricas, como ferramenta matemática para modelar e a analisar a qualidade de produto de forma multidimensional
- Pesquisa bibliográfica sobre o arcabouço de métodos, procedimentos e ferramentas utilizado em trabalhos relacionados
- Desenvolvimento de um protótipo funcional, com uso de dados reais e simulados que permitiu explorar, de forma preliminar, a viabilidade computacional da ideia proposta

Conclusões

Macroatividades Realizadas

- Publicação de um artigo sobre as ideias iniciais desta proposta, que foram recentemente discutidas no ESEM.
 - Neri, Hilmer Rodrigues; Travassos, Guilherme Horta. Measuresoftgram. In: the 12th ACM/IEEE International Symposium, 2018, Oulu. Proceedings of the 12th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement - ESEM '18. New York: ACM Press, 2018. p. 1-4. <http://dx.doi.org/10.1145/3239235.3267438>

Conclusões

Macroatividades Previstas

- Formalizar o modelo matemático
- Investigar a eficiência das técnicas de decomposição de tensores nesse contexto de dados da qualidade de produto de software, com objetivo principal de identificar relações latentes entre esses dados, de forma multidimensional;
- Substituir o *back-end* Numpy pelo TensorFlow
- Investigar se os tratamentos de esparsidade providos pela decomposição dos tensores podem ser aplicados de forma eficiente a dados de qualidade de produto de software
- Aprofundar o tratamento para o valor zero, 0, no modelo proposto
- Avaliar se a medida de similaridade de vetores se mostra adequada para interpretação para a diferença percebida na qualidade de produto planejada e realizada
- Instrumentação de um ambiente de experimentação contínua aplicado ao contexto da Administração Pública Federal Brasileira
- Definir o conjunto de organizações públicas participantes dos estudos experimentais a serem conduzidos para investigação, no contexto da indústria, sobre as ideias desta pesquisa
- Redigir a Tese de Doutorado e defendê-la na presença de uma banca

Conclusões

Veículos candidatos a submissão dos resultados desta pesquisa

- Submissão e publicação de artigos em eventos e periódicos.
Algumas sugestões:
 - Conferências
 - International Workshop on Software Measurement (IWSM)
 - International Conference on Software Process and Product Measurement (MENSURA)
 - International Conference on Software Engineering (ICSE)
 - Simpósio Brasileiro de Engenharia de Software (SBES/CBSoft)
 - Workshop Brasileiro de Qualidade de Software (SBQS)
 - Periódicos
 - Software Quality Journal
 - Empirical Software Engineering
 - Information and Software Technology
 - Transaction Software Engineering
- E por fim, a Tese de Doutorado, em si.

Conclusões

Contribuições Previstas

- Um conjunto de evidências (positivas ou negativas) sobre o uso de tensores como ferramenta para modelar e analisar multidimensionalmente a qualidade de produto e em uso de software;
- O uso de experimentação contínua para apoiar a tomada de decisão sobre a aceitação de releases, com respeito a qualidade em uso, a partir de evidências da qualidade de produto;
- Apoiar o processo de gerenciamento da qualidade de produto de software por parte da Administração Pública Federal Brasileira. Em particular a atividade ateste técnico e negocial por parte dos gestores públicos nas contratações de serviços de desenvolvimento de software;
- Uma base de conhecimento sobre a análise multidimensional de características de qualidade do software, além do uso de experimentação contínua no contexto da Administração Pública Federal.
- Um conjunto de estudos experimentais planejados e executados com o objetivo de avaliar esta proposta
- A disponibilização de uma infraestrutura computacional de experimentação contínua que apoie de forma sistemática o ateste de versões de produtos de software com respeito a qualidade de sistemas de software

Conclusões

Finais

- Foi apresentada nesta proposta, para Exame de Qualificação de Doutorado, nossa visão do futuro de apoiar a observação contínua e multidimensional da qualidade do produto de software em ambientes contínuos de software. Foi chamado MeasureSoftGram.
- Nossa motivação para lidar com esse problema reside no contexto da Contratação Brasileira de Desenvolvimento de Software Público, que opera uma quantidade considerável de recursos públicos em diferentes projetos de software anualmente. Aproximadamente algo em torno de 4 bilhões de reais/ano, somente na esfera federal
- Portanto, melhorar a tomada de decisão sobre a aceitação de lançamentos de produtos de software é vital para proteger os recursos públicos e garantir que produtos de software de alta qualidade (sob a perspectiva das partes interessadas) estejam disponíveis para a sociedade brasileira.