



AARM

Conectando Arquitetura de Software ao Negócio

Uma introdução ao **Agile Architecture Risk Management** e como ele transforma decisões técnicas em vantagem competitiva.

Criado por Cristiano Gomes.

Adotado e recomendado pelo Software Engineering Institute / Carnegie Mellon University.

O PROBLEMA

Como a Arquitetura se perdeu?

- ◆ O modelo **Waterfall** não existe mais, mas muitos times de Arquitetura ainda operam como se existisse
- ◆ **Distanciamento** das squads: a Arquitetura decide longe de quem constrói
- ◆ **Alheios** aos problemas de produção: quem desenha raramente sente o incidente
- ◆ Um novo jogo exige **uma nova estratégia**, não um comitê mais rígido

“Menos problemas em produção, menos salas de guerra.”

A premissa por trás do AARM: problemas de design descobertos tarde no ciclo de vida são os mais caros de corrigir. Gerenciar risco de arquitetura cedo, e continuamente, é mais barato que qualquer sala de guerra.

DE VOLTA AOS PRINCÍPIOS

O manifesto ágil mostra o caminho

“Continuous attention to technical excellence and good design enhances agility.”

Principles behind the Agile Manifesto

“The best architectures, requirements, and designs emerge from self-organizing teams.”

Principles behind the Agile Manifesto

O AARM não inventa um valor novo, ele operacionaliza estes dois princípios num método que o time inteiro consegue rodar.

O FRAMEWORK

O que é o AARM

Agile Architecture Risk Management é um framework para gerenciar riscos de arquitetura em produtos digitais, conectando cada decisão técnica aos objetivos estratégicos do negócio, em quatro etapas contínuas.

- ◆ **Aberto:** documentação, templates e método públicos
- ◆ **Criado em 2022:** por Cristiano Gomes, a partir de prática em grandes operações
- ◆ **Validado em 2026:** adotado pelo SEI / Carnegie Mellon University em relatório oficial
- ◆ **Leve:** roda em sessões curtas, com o time que você já tem

AS 4 ETAPAS

- ◆ 1. Estratégia do produto
- ◆ 2. Características de arquitetura
- ◆ 3. Risk-storming
- ◆ 4. Histórias de arquitetura

O CICLO

Quatro etapas, um ciclo contínuo

Estratégia do produto

Objetivos confirmados com o negócio



Características de arquitetura

*ilities priorizadas: top 3 de 7



Risk-storming

Riscos identificados por todo o time



Histórias de arquitetura

backlog priorizado. Codar a Arquitetura



mais adaptativo



mais ágil



Quanto mais perto do código, maior a cadência: estratégia se revisita de tempos em tempos; risk-storming se repete a cada ciclo.

AS ETAPAS

Cada etapa responde uma pergunta e produz um artefato

ETAPA 1

Estratégia do produto

Quais objetivos de negócio a arquitetura precisa habilitar?

→ 3-5 objetivos confirmados

ETAPA 2

Características de arquitetura

O que a plataforma precisa ser para habilitá-los?

→ Top 3 de 7 ilities + trade-offs

ETAPA 3

Risk-storming

O que ameaça essas características, onde e com que gravidade?

→ Mapa de riscos priorizado

ETAPA 4

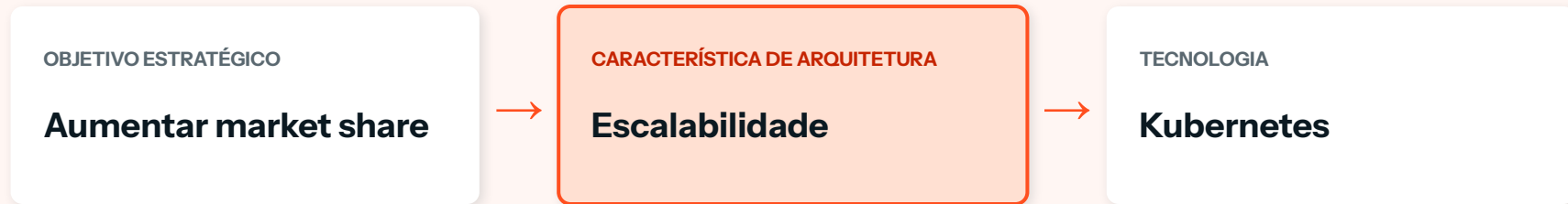
Histórias de arquitetura

O que fazer a respeito, em que ordem e por quê?

→ Backlog com linhagem completa

CONECTANDO NEGÓCIO E TECNOLOGIA

A tecnologia é a última a entrar na frase



*“Para habilitar o crescimento de **market share**, nossa plataforma será **escalável**. Kubernetes será uma das tecnologias adotadas para habilitar esse objetivo estratégico do negócio.”*

É esse formato, a história de arquitetura, que faz um item técnico ser priorizável por qualquer stakeholder.

A TÉCNICA MAIS TANGÍVEL

Risk-storming em uma sessão

- 1 **Diagramas na parede:** um ou mais C4 mostrando o que será construído ou alterado
- 2 **Identificação silenciosa:** cada pessoa escreve seus riscos, sem discussão e o time sênior fala por último
- 3 **Riscos no mapa:** post-its colados sobre a área do diagrama; aglomeração = ponto quente
- 4 **Revisão e síntese:** atenção especial aos riscos solitários e às divergências de nota

Impacto × Probabilidade

	1	2	3
Impacto	2	4	6
	3	6	9
	Probabilidade		

Técnica original de Simon Brown (riskstorming.com), incorporada pelo AARM como etapa 3 do ciclo.

POR QUE CONFIAR

Validado por quem mais entende de risco de arquitetura

SEI REPORT · MARÇO 2026

Managing Architectural Risk During Agile Development

O Software Engineering Institute da Carnegie Mellon University adotou o AARM como metodologia de referência para avaliar riscos de arquitetura no desenvolvimento ágil.

sei.cmu.edu

“This is a well thought out and structured approach, and includes all of the elements I use when analyzing architectural risk.”

Mark Richards

Autor best-seller em arquitetura de software

Praticado em operações como **Natura · AB InBev · Raízen · Departamento de Defesa dos EUA · Moviq**

QUANDO PARAR DE RODAR O AARM

Nunca.

- ◆ Revise as características priorizadas periodicamente. E reconsidere as não priorizadas.
- ◆ Avalie SLAs, SLOs e métricas DevOps a cada ciclo.
- ◆ Meça e acompanhe a composição do débito técnico.
- ◆ Confirme os objetivos de negócio: se eles mudaram, o ciclo recomeça.

PROPOSTA

Como começar aqui: um ciclo piloto

1

Escolher um produto

Um único produto ou fluxo de valor, de preferência o que mais dói hoje.

2

Rodar o primeiro ciclo

Quatro sessões curtas, uma por etapa, com negócio, produto e time técnico. Tudo que precisamos é gratuito: templates e guias em <https://elegantgarden.software/aarm>

3

Levar ao backlog e medir

Histórias de arquitetura priorizadas com produto, e um antes/depois para avaliar o método com dados.

Alternativa com facilitação externa: **o AARM Sprint** roda o ciclo completo em 4 semanas com os criadores do framework.

Acesse: elegantgarden.software/aarm/aarm-consultoria.html

Vamos colocar a Arquitetura de volta no jogo?

- ◆ Documentação completa do framework
- ◆ Templates gratuitos (canvas, matriz, board de risk-storming)
- ◆ White paper e SEI Report

Acesse: elegantgarden.software/aarm