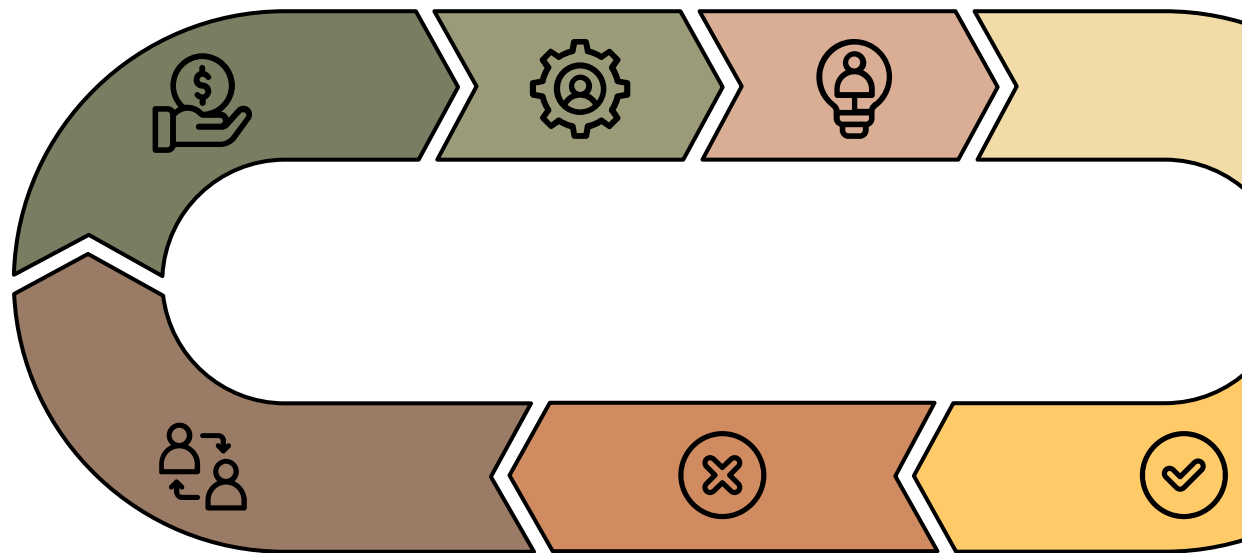


Projeto de Bloco: Desenvolvimento Front-end com Frameworks



Instituto Infnet

Prof. Kennedy Carvalho

Olá {{mundo}} !

Kennedy Carvalho

A.k.a

Engenheiro de Software

Who?

@kndrio
<https://knd.rio>



Sobre a Disciplina

Objetivo da Disciplina

Desenvolver um projeto que una as competências apresentadas nas disciplinas regulares

Tópicos da Aula

Definições

Ciclo de Vida

Metodologias

Boas Práticas

Scrum

Definições

IEEE - Engenharia de Software

Aplicação de um processo sistemático, disciplinado e quantificado ao desenvolvimento, a operação e a manutenção de software.

Pressman - Processo de software

Metodologia para as atividades, ações e tarefas necessárias para desenvolver um software de alta qualidade.

Ciclo de Vida do Software

Sommerville

Especificação

A funcionalidade do software e as restrições a seu funcionamento devem ser definidas.

Projeto e Implementação

O software deve ser produzido para atender às especificações

Validação

O software deve ser validado para garantir que atenda às demandas do cliente

Evolução

O software deve evoluir para atender às necessidades de mudança dos clientes.

Motivos para desenvolver um software



Redução de custos.



Aumento do nível de serviço e desempenho dos recursos humanos.



Melhoria do processo de tomada de decisões pela administração da empresa.



Aumento da vantagem competitiva sobre os concorrentes.



Modelo de negócio onde o software é o principal produto da empresa.



Aumento da qualidade dos produtos e serviços, por meio da automação das rotinas.



Importância da Metodologia



Falta de qualidade do produto final;



Falhas na comunicação



Não cumprimento dos prazos;



Não cumprimento dos custos.



Gerenciamento de complexidade;



Comunicação entre pessoas envolvidas;



Redução dos custos no desenvolvimento;



Previsão do comportamento futuro dos sistemas.



Primeiras Metodologias

Os primeiros processos de desenvolvimento de software foram propostos ainda nas décadas de 60 e 70 — eram estritamente sequenciais.

Essa primeira visão de processo era natural, visto que projetos de Engenharia tradicional também são sequenciais e precedidos de um planejamento detalhado.

Começou-se a perceber que software é diferente de outros produtos de Engenharia.

Essa percepção foi ficando clara devido aos problemas frequentes enfrentados por projetos de software nas décadas de 70 a 90.

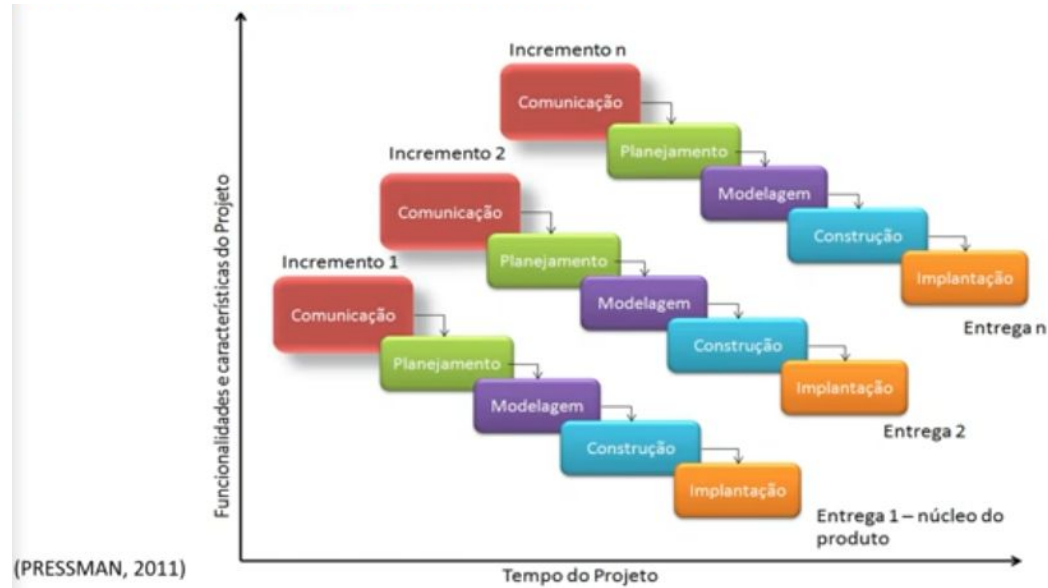


Modelo em Cascata

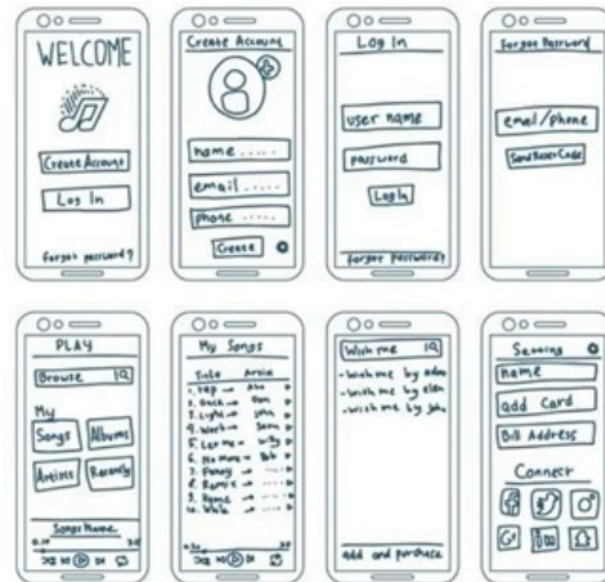


(SOMMERVILLE, 2011)

Modelo Incremental

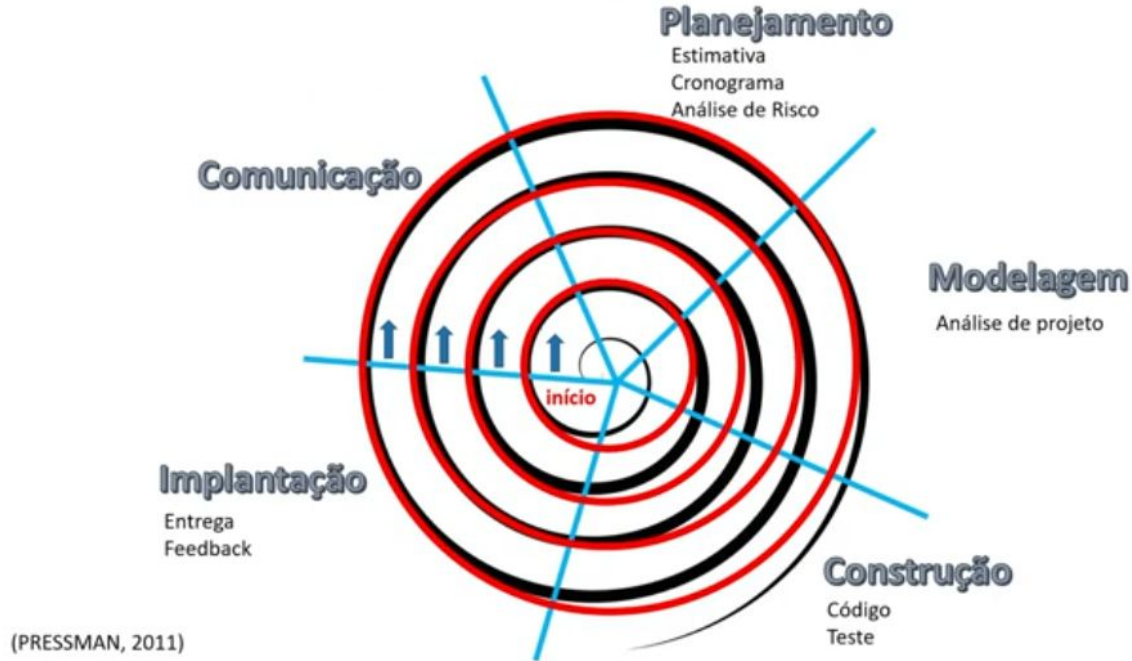


Modelo de Prototipação



(PRESSMAN, 2011)

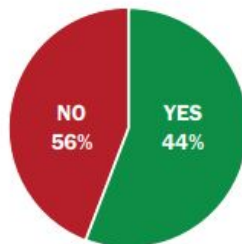
Modelo Espiral ou Evolutivo





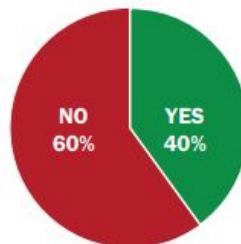
CHAOS REPORT 2015

ONBUDGET



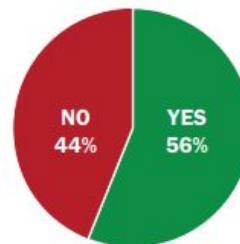
The percentage of projects that were OnBudget from FY2011–2015 within the new CHAOS database.

ONTIME



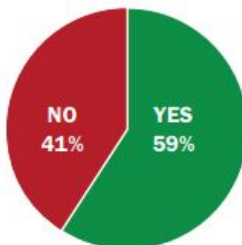
The percentage of projects that were OnTime from FY2011–2015 within the new CHAOS database.

ONTARGET



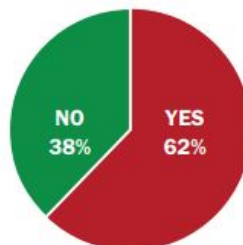
The percentage of projects that were OnTarget from FY2011–2015 within the new CHAOS database.

VALUABLE



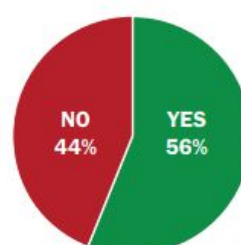
The percentage of projects considered valuable from FY2011–2015 within the new CHAOS database.

ONGOAL



The percentage of projects that were OnGoal from FY2011–2015 within the new CHAOS database.

SATISFACTORY



The percentage of projects considered satisfactory from FY2011–2015 within the new CHAOS database.

Metodologias Ágeis

Através deste trabalho, passamos a valorizar:

Indivíduos e interações mais que processos e ferramentas

Software em funcionamento mais que documentação abrangente

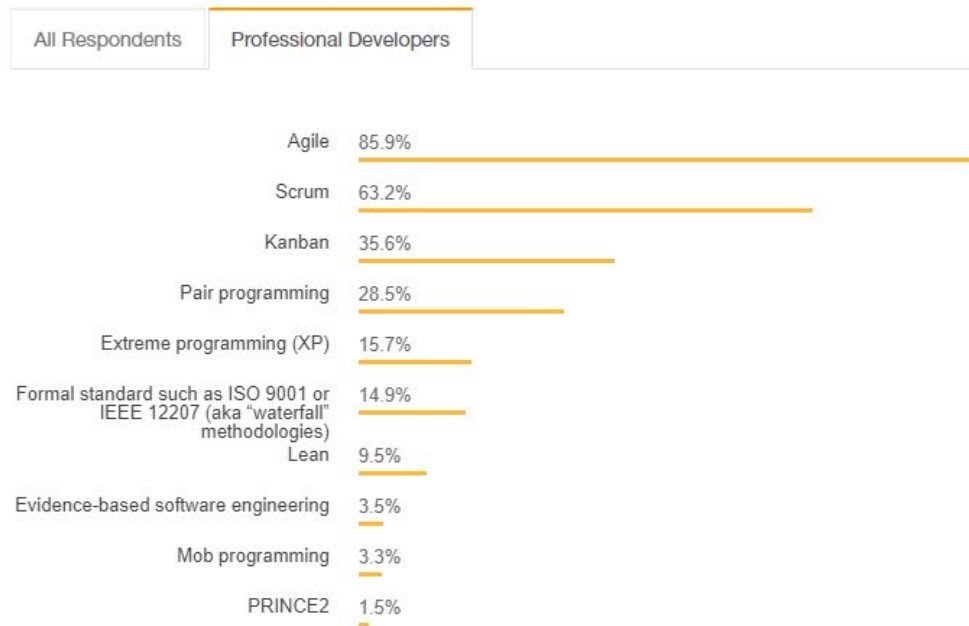
Colaboração com o cliente mais que negociação de contratos

Responder a mudanças mais que seguir um plano

<https://agilemanifesto.org/>

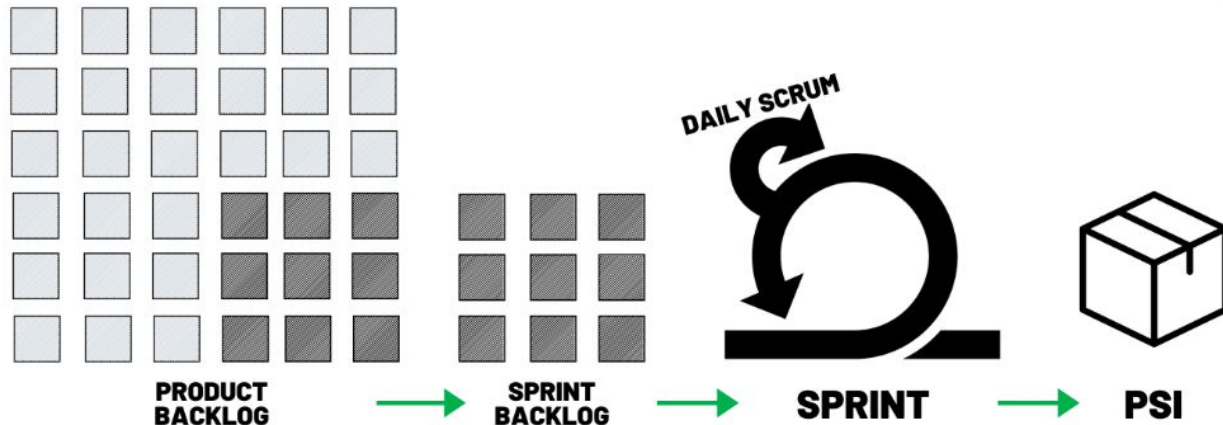
Metodologias Ágeis

Which Methodologies Do Developers Use?



57,075 responses; select all that apply

SCRUM CHEAT SHEET



ROLES

PRODUCT OWNER

- Develops product vision
- Owns product backlog
- Prioritizes backlog items

SCRUM MASTER

- Process coach
- Removes impediments
- Facilitates team meetings

THE TEAM

- Cross-functional
- Self-organizing
- Has skills to complete the sprint work

ARTIFACTS

PRODUCT BACKLOG

- Lists all the work on a product or project
- Never complete, always changing
- Higher order = higher priority

SPRINT BACKLOG

- Lists work to be done in the current sprint
- Pulled from product backlog
- Items are broken into tasks

SHIPPABLE INCREMENT

- Potentially shippable product increment
- Completely designed, coded, and tested
- Meets all acceptance criteria

ACTIVITIES

SPRINT PLANNING

- Plan created for what is to be delivered in the upcoming sprint.

THE SPRINT

- Consistent duration of time where the team completes work.

DAILY SCRUM

- 15 minute meeting where team reports:
 - 1.What did you do yesterday?
 - 2.What are you doing today?
 - 3.List any impediments.

SPRINT REVIEW

- Team shows off completed work to Product Owner

SPRINT RETRO

- Team discusses how the sprint went and improvements to make next sprint



agile
meridian

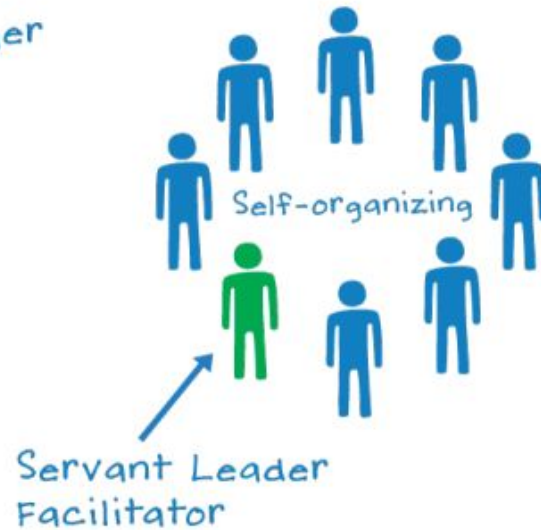
AGILEMERIDIAN.COM

Papel do Scrum Master

Traditional Teams



Agile Teams



o que faz um Scrum Master?

remove impedimentos

mantém a comunicação fluindo

mantém foco das reuniões

conhece o Scrum

agiliza comunicação com outras equipes (BDFTP, infra, etc)

resiste às adversidades

cobra horários do Customer Team

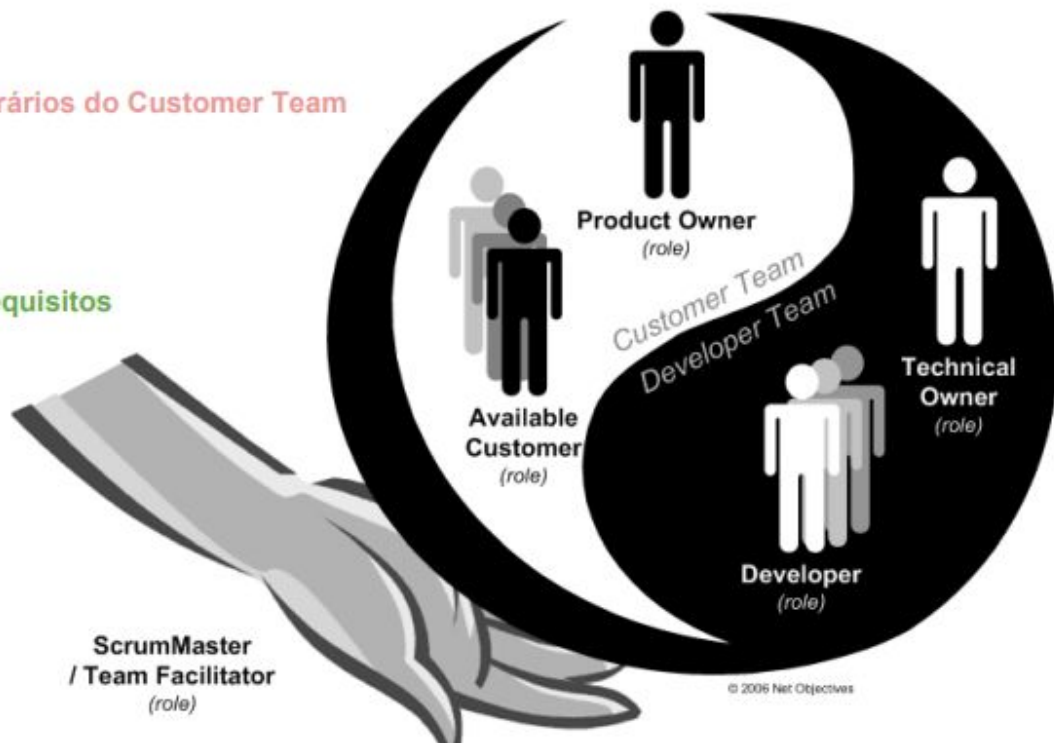
é comprometido

conhece os requisitos

escala problemas

mantém cronograma

reporta status



O que o Scrum Master NÃO deve ser:

decorador de quadro

Visualize		Limit Work in Progress		Manage the Flow		Make Process Policies Explicit		Improve Collaboratively using Mo.	
Visualize Work	Visualize workflow	Limit Work in Progress using Lane Limits	Full Scheduling	Track Cycle Times	Manage work in front of the board (e.g. Daily Sync)	Make CURRENT policies explicit	Evolves policies to overcome obstacles and advance towards Capabilities Goal	Start with what you do now	Initially, respect current roles, responsibilities & job titles
Use a Physical Card Wall	Use an electronic kanban system	Smaller Batches	Allocate WIP Capacity to Classes of Service	Establish Cadence - Planning, Delivery, Retrospection	Cumulative Flow Diagram (CFD)	Establish Definition of Done for each Stage	Establish "how to Pull work" policies	Agree to pursue incremental, evolutionary change	Use Flow principles to improve
Create flow-oriented views on your existing work mgmt system	Visualize Buckets/Issues/Imps	Stop Starting Start Finishing (e.g. hard WIP limits)		Monitor Cycle Time Control Charts	Track Process Cycle Efficiency (aka Flow Efficiency)	Establish Classes of Service to deal with variety of work coming in	Establish Definition of READY for each Stage	Use Toyota Kata to improve	Use A3 thinking to coach people towards improvement on key obstacles to
Design board using "The work knows"	Design board using "Start with the Flow"			Establish Service Level Expectations/Forecast for work groupings	Release Dump	Setup "Work Size Limit" policies as part of definition of READY	Decide UP FRONT how to deal with expedited demand	Use Theory of Constraints Five focusing steps to deal with constraints/bottlenecks	Track reasons for Rework and focus on top reasons
Emphasize flow-moving items in real time (e.g. Zombie Cycle Times)	Emphasize Network			Track bigger projects using trends on CFDs		Make FIXED DATE work but discourage and limit its use	Encourage "Housekeeping"/"Am" as a managed class of work	Track reasons for long cycle time exceptions and deal with top reasons	Retrospectives
Visualize multiple levels using Expand/Collapse	Kanban Networks e.g. Product Streams + Technical Teams + Big Picture					"Stop the line"	Establish Estimation policies	Operations Review	Improvement Work Kanban System/Stream
						Use Scrumdan policies		Look for 8 Wastes	Use Five Whys systematically and iteratively



Para o Product Owner

- 1) Claramente comunicar a visão, objetivo e itens do Backlog do Produto para a Equipe de Desenvolvimento (Não é proxy!!);
- 2) Compreender a longo-prazo o planejamento do Produto no ambiente empírico;
- 3) Compreender e praticar a agilidade;

Para a Equipe de Desenvolvimento

- 1) Remover impedimentos para o progresso da Equipe de Desenvolvimento;
- 2) Facilitar os eventos Scrum conforme exigidos ou necessários;



Ready? Prontos?