



# Software Engineering Method and Theory

Geraldo Xexéo

DCC/IM & PESC/COPPE

Universidade Federal do Rio de Janeiro

xexeo@cos.ufrj.br

<http://www.cos.ufrj.br/~xexeo>

<http://www.xexeo.net/>

# Situação Atual

- Existe muito conhecimento sobre como desenvolver software
- Trocamos de método de desenvolvimento sem entender as consequências
  - Modismos
  - O que virá a seguir?
- Não há um acordo na comunidade em quais são as práticas adequadas para o desenvolvimento de software

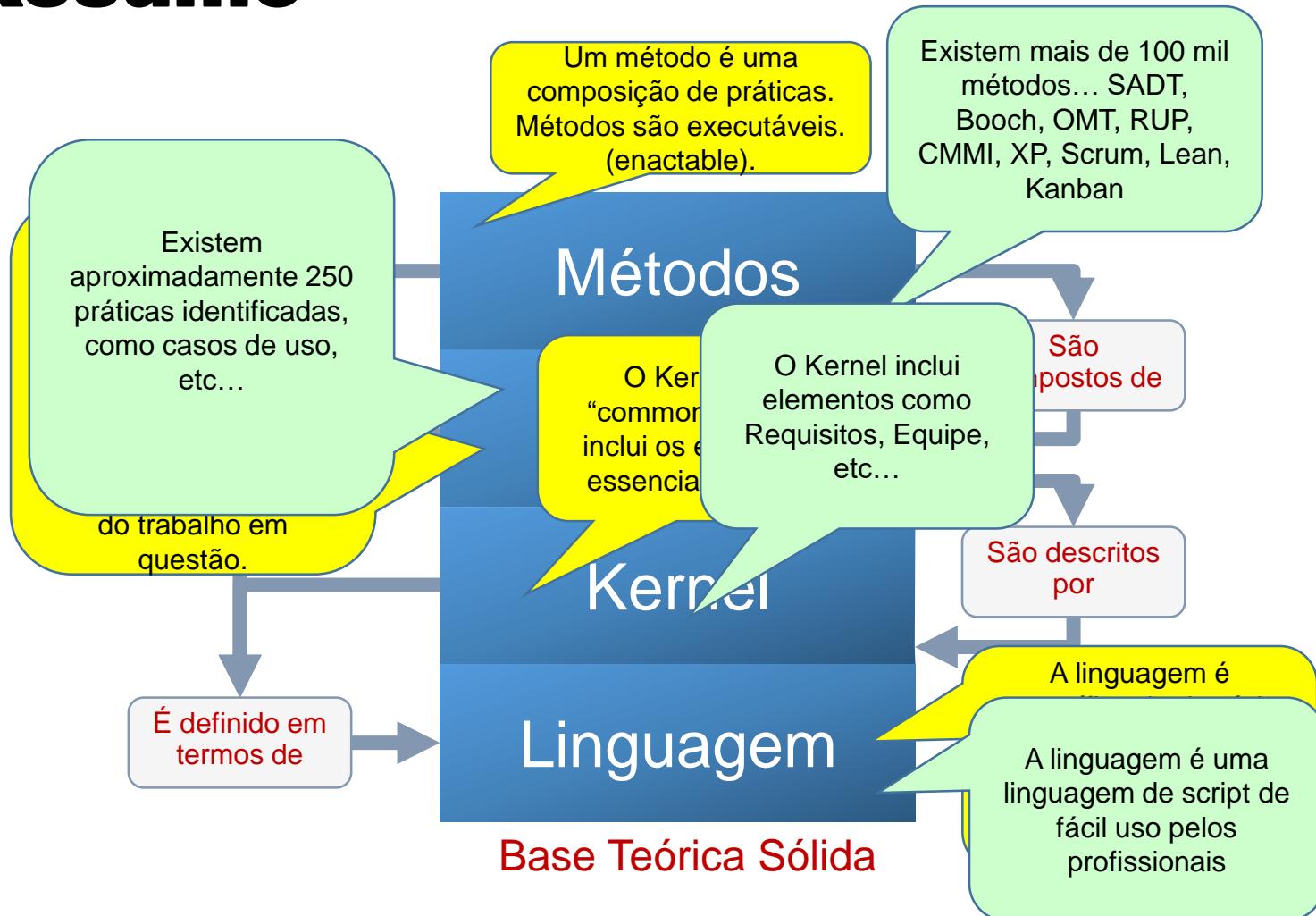
# Práticas Imaturas

- Prevalência de modismos
- Falta de uma base teórica sólida
- Grande número de métodos e suas variantes
  - Dificuldade de entender as diferenças
  - Diferenças aumentadas artificialmente
- Falta de avaliação experimento e validação
- Academia e indústria divididas

# **SEMAT**

- Refundar a Engenharia de Software com base em
  - Uma teoria sólida
  - Princípios comprovados
  - Melhores Práticas
- Buscar Inicialmente
  - Base teórica sólida
  - “Kernel” de elementos amplamente aceitos

# Resumo

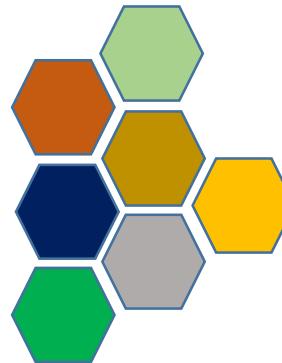
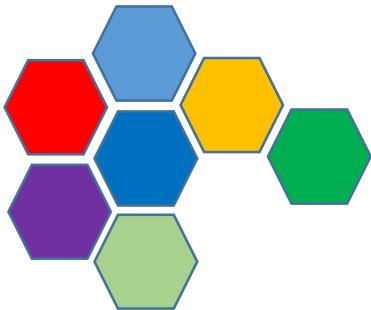


# O Kernel

- Alphas
  - Representações das coisas essenciais com que trabalhamos
  - Descrições das coisas que o time gerênciа, produz e usa no processo de desenvolver, manter e suportar um bom software.
  - Âncoras para sub-alphas adicionais e produtos de trabalho exigidos pelas práticas de engenharia de software
- Espaços de atividade
  - Representações das coisas essenciais a fazer.
    - Fornecem descrições dos desafios que os times encontram enquanto desenvolvendo, mantendo e suportando sistemas de software, e
    - tipos de coisas que os times vão fazer para atingi-los.
- Competências
  - Habilidades Necessárias

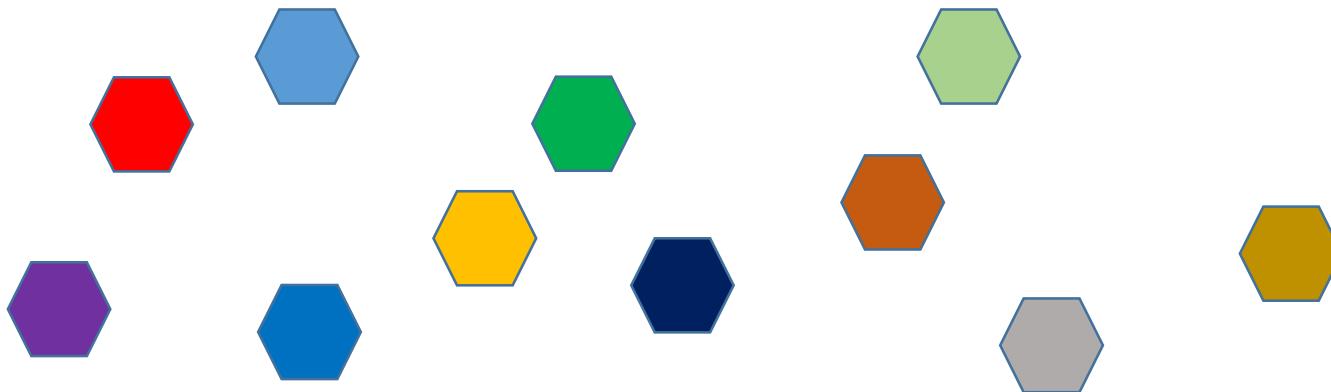
# Método

- Uma composição de Práticas
- São executáveis
  - Enactable
- Mais de 100.000 encontrados
  - Incluindo SADT, RUP, Booch, OMT, CMMI, XP, Scrum, Lean, Kanban



# Prática

- Uma abordagem repetitível para fazer algo com um objetivo específico em mente
- Fornece uma maneira sistemática e verificável de abordar um aspect particular do trabalho sendo feito
- Pelo menos 250 identificadas
  - Casos de Uso, Histórias do Usuário, Componentes, etc...



# **Kernel**

- É um campo comum que inclui os elementos essenciais da Engenharia de Software
- Elementos essenciais incluem:
  - Partes interessados, oportunidades, equipe,...

# Linguagem

- Definem métodos, práticas e os elementos essenciais do Kernel
- Define as coisas essenciais para as quais um projeto de desenvolvimento precisa prestar atenção
- Linguagem de Script

# O Kernel Essencial

- Define uma base comum para definição de prática de desenvolvimento de software
  - De forma que possam ser definidas e aplicadas independentemente
- Estruturado em 3 áreas
  - Cliente
    - Contém tudo relacionado com o uso e exploração do software sendo produzido
  - Solução
    - Contém tudo em relação a especificação e desenvolvimento do sistema de software
  - Empreendimento
    - Contém tudo em relação ao time e a forma como aborda o trabalho



# O Kernel Essencial (II)

- Cada área contém
  - Alphas
  - Espaços de Atividades
  - Competências
- As áreas fornecem
  - Coisas essenciais a serem produzidas, gerenciadas e controladas
  - Coisas essenciais a serem feitas
  - Competências essenciais

# **Kernel: Áreas de Interesse**

- Cliente
  - Tudo que tem relação com o uso e exploração do sistema de software sendo produzido
- Solução
  - Tudo para realmente usar o software
- Empreendimento
  - Tudo que tem a ver com o trabalho

**Cliente**

**Solução**

**Empreendimento**

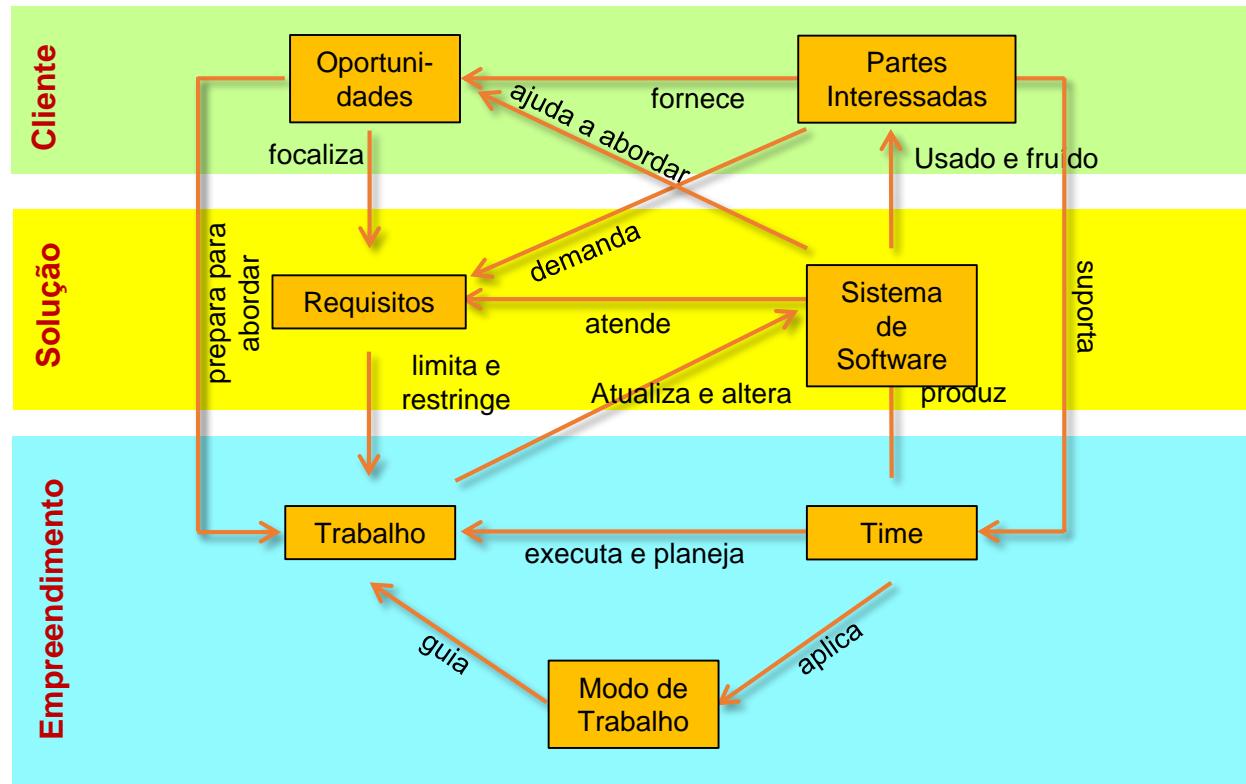
# Alphas

$\alpha$

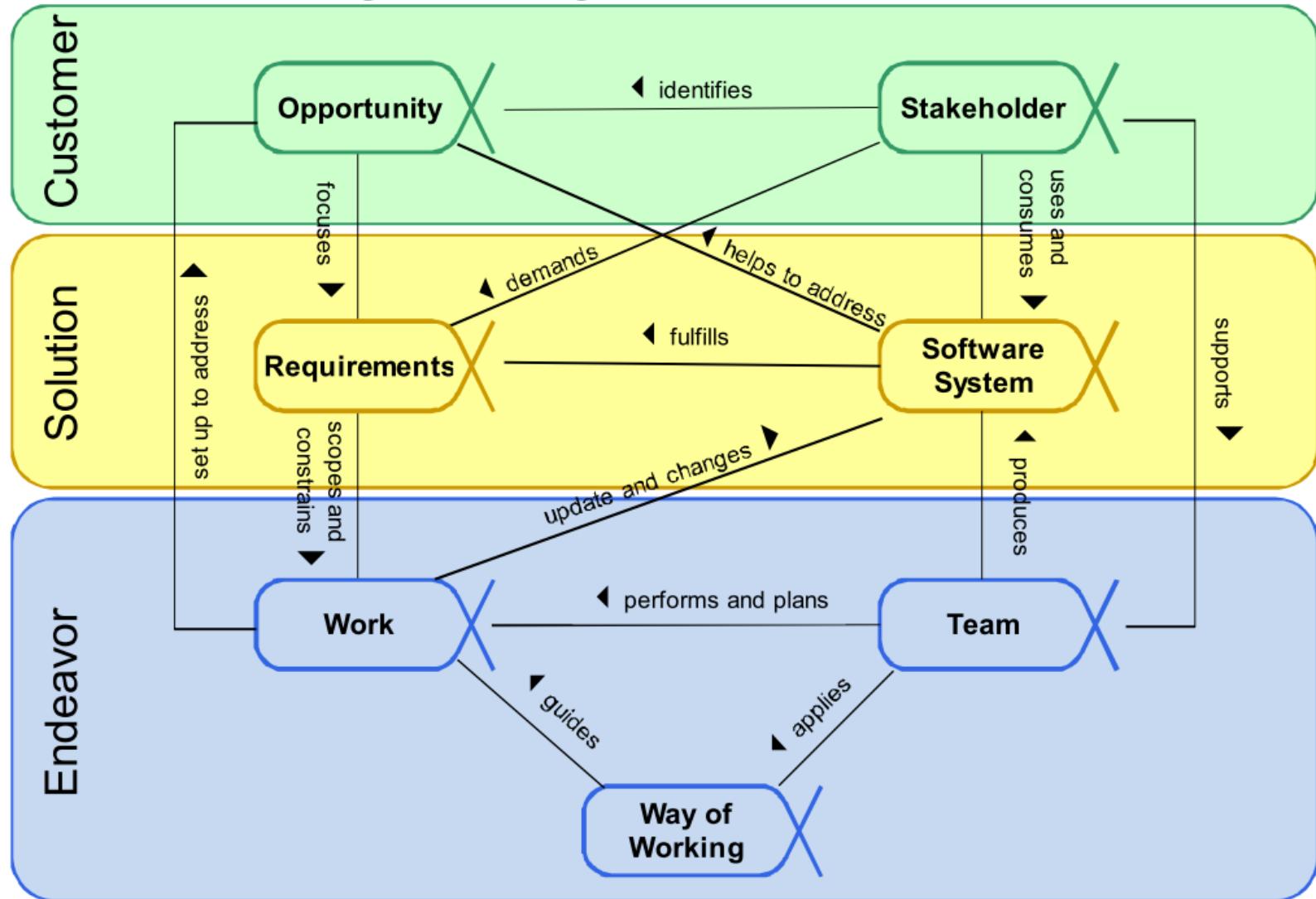
- Representações das coisas essenciais com que trabalhamos
- Descrições das coisas que o time gerência, produz e usa no processo de desenvolver, manter e suportar um bom software.
- Âncoras para sub-alphas adicionais e produtos de trabalho exigidos pelas práticas de engenharia de software
- Capturam conceitos chave da ES
- Permitem rastrear e avaliar a saúde e o progresso de qualquer empreendimento de ES
- Fornecem uma base comum para a definição de práticas e métodos da ES

# Alphas

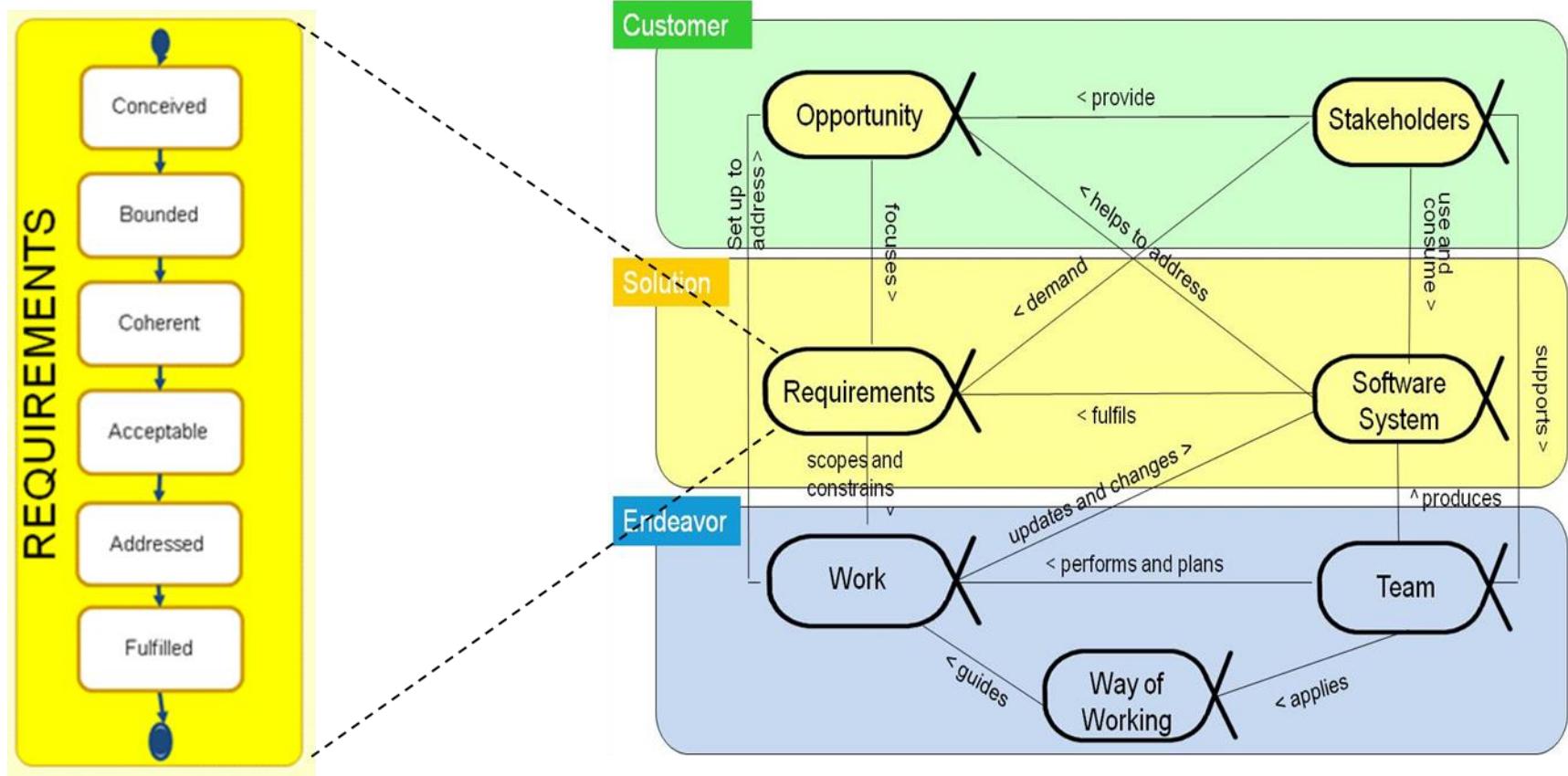
Práticas e ferramentas escolhidas sob medida e utilizados pelo time para guiar e apoiar seu trabalho. O time desenvolve seu MdTjunto com o entendimento de sua missão e do seu ambiente de trabalho. No decorrer do trabalho o time continuamente reflete sobre seu modo de trabalho e o adapta como for necessário para o contexto correto.



# O Kernel - Alphas



# Detalhamento em Estados



# Cliente

- Partes Interessadas

- Pessoas, grupos ou organizações que afetam ou são afetadas pelo sistema de software. As P.I. fornecem oportunidades e são a fonte de requisitos e financiamento do sistema de software. Elas precisam ser envolvidas atrás do empreendimento para apoiar o time e garantir que um sistema de software aceitável é produzido.

- Oportunidades

- Circunstâncias que tornam apropriado o desen. ou mudança de um sistema de software. Articula a razão da criação de um sistema novo ou modificado. Representa o entendimento compartilhado pelo time sobre as necessidades das P.I., ajudam a formar os requisitos fornecendo justificativas para seu desenvolvimento.

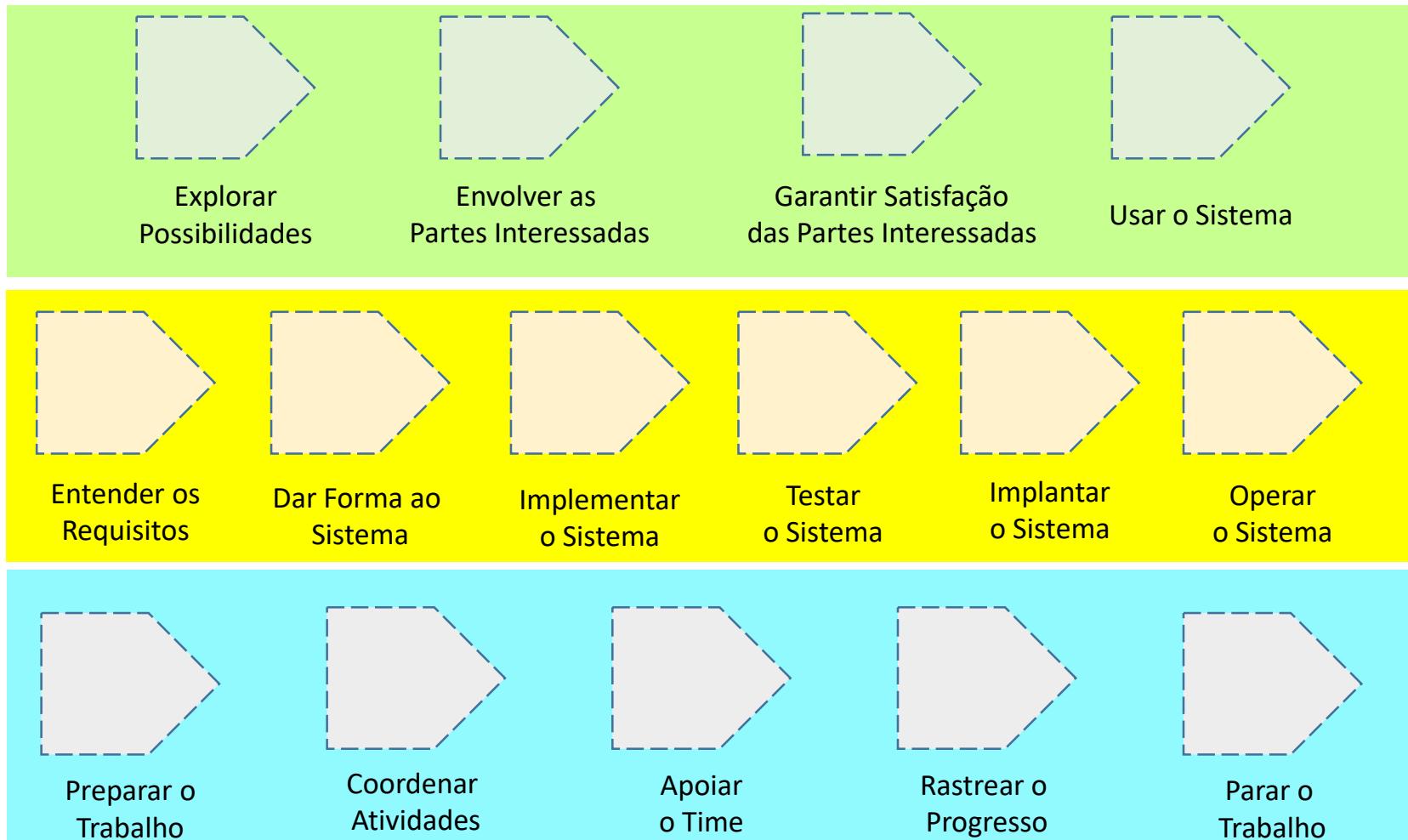
# Solução

- Requisitos
  - O que o software precisa fazer para atender as oportunidades e satisfazer as P.I.. É importante descobrir o que é necessário para o SdS, compartilhar esse entendimento entre as P.I. e o time e usar para dirigir o desenvolvimento e teste do sistema
- Sistema de Software (SdS)
  - Um sistema feito de software, hardware e dados que fornece valor primário pela execução de software. O produto primário de qualquer empreendimento de engenharia de software, um SdS pode ser parte de um software maior, hardware ou solução de negócio.

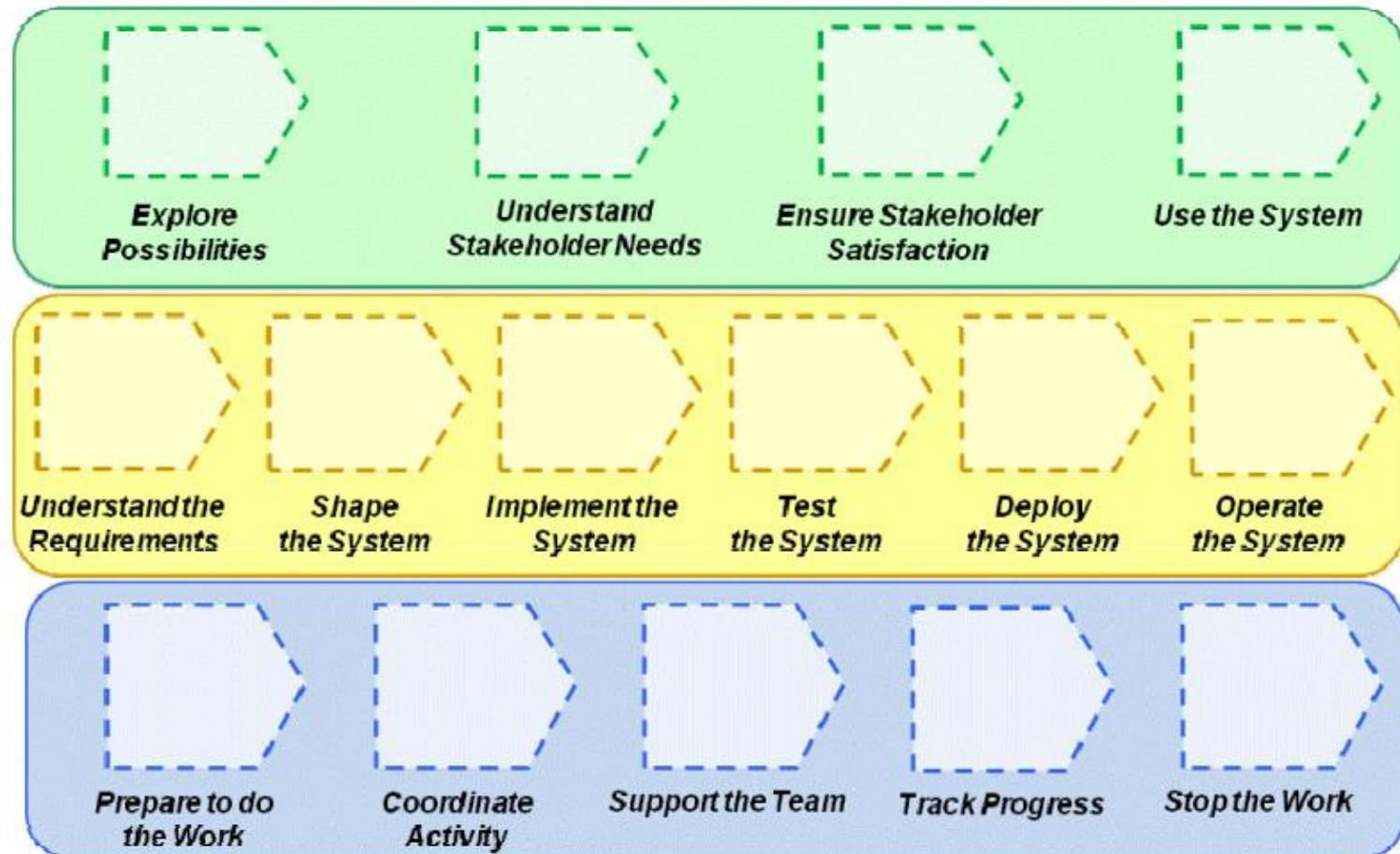
# **Empreendimento**

- Time
  - Um grupo de pessoas ativamente engajadas no desenvolvimento, manutenção, entrega e suporte de um SdS específico. O time planeja e realiza o trabalho feito para atualizar e mudar o SdS
- Trabalho
  - Atividade envolvendo esforço físico ou mental feito com o propósito de alcançar um resultado. No contexto da ES, trabalho é tudo que o time faz para atingir seus objetivos e de produzir um SdS atendendo aos Req. e endereçando as Oport. Apresentadas pelo cliente. O trabalho é guiado por práticas que fazem o modo de trabalho.
- Modo de Trabalho
  - Práticas e ferramenta escolhidos sob medida e utilizados pelo time para guiar e apoiar seu trabalho. O time desenvolve seu MdTjunto com o entendimento de sua missão e do seu ambiente de trabalho. No decorrer do trabalho o time continuamente reflete sobre seu modo de trabalho e o adapta como for necessário para o contexto corrente.

# Kernel: Espaços de Atividades



# Kernel: Espaços de Atividades

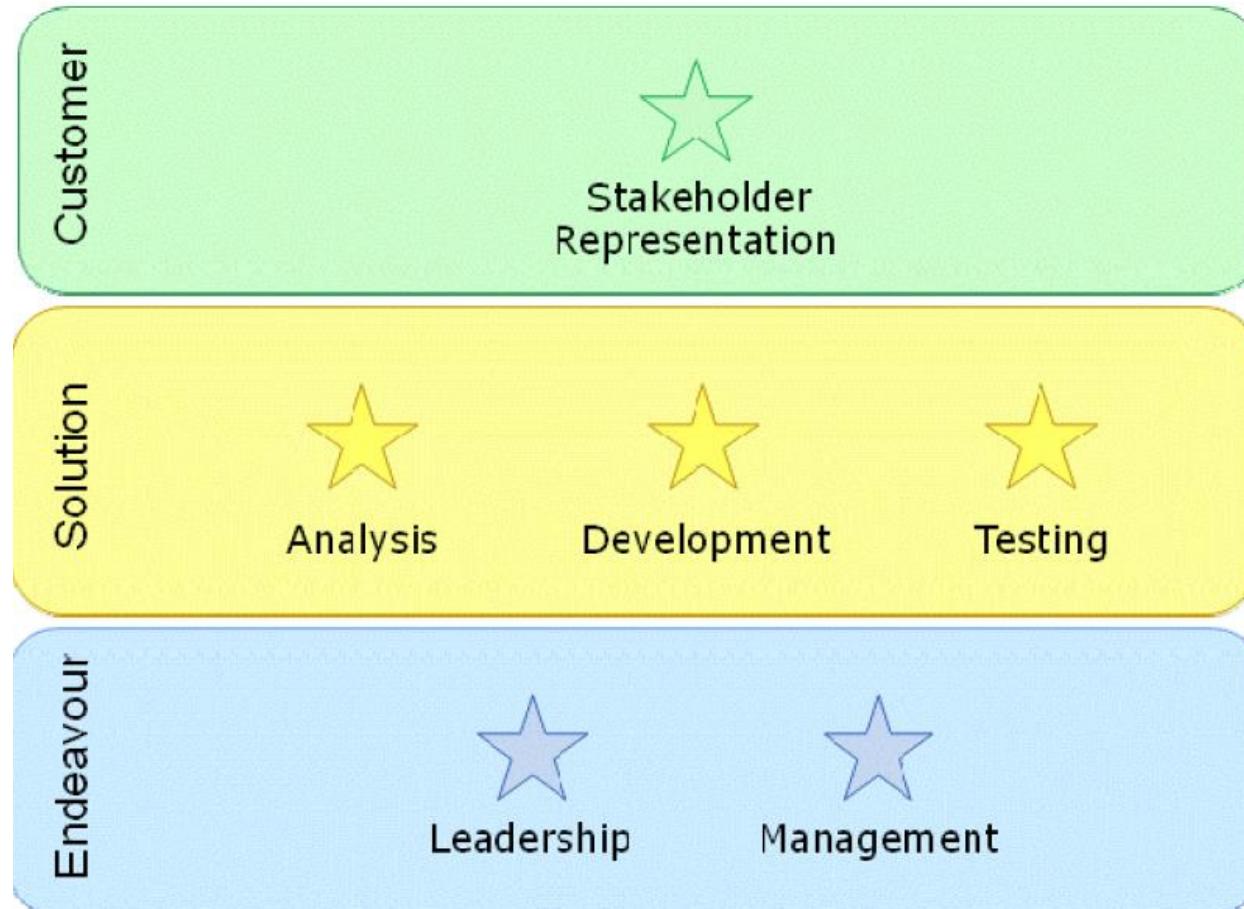


# Competências

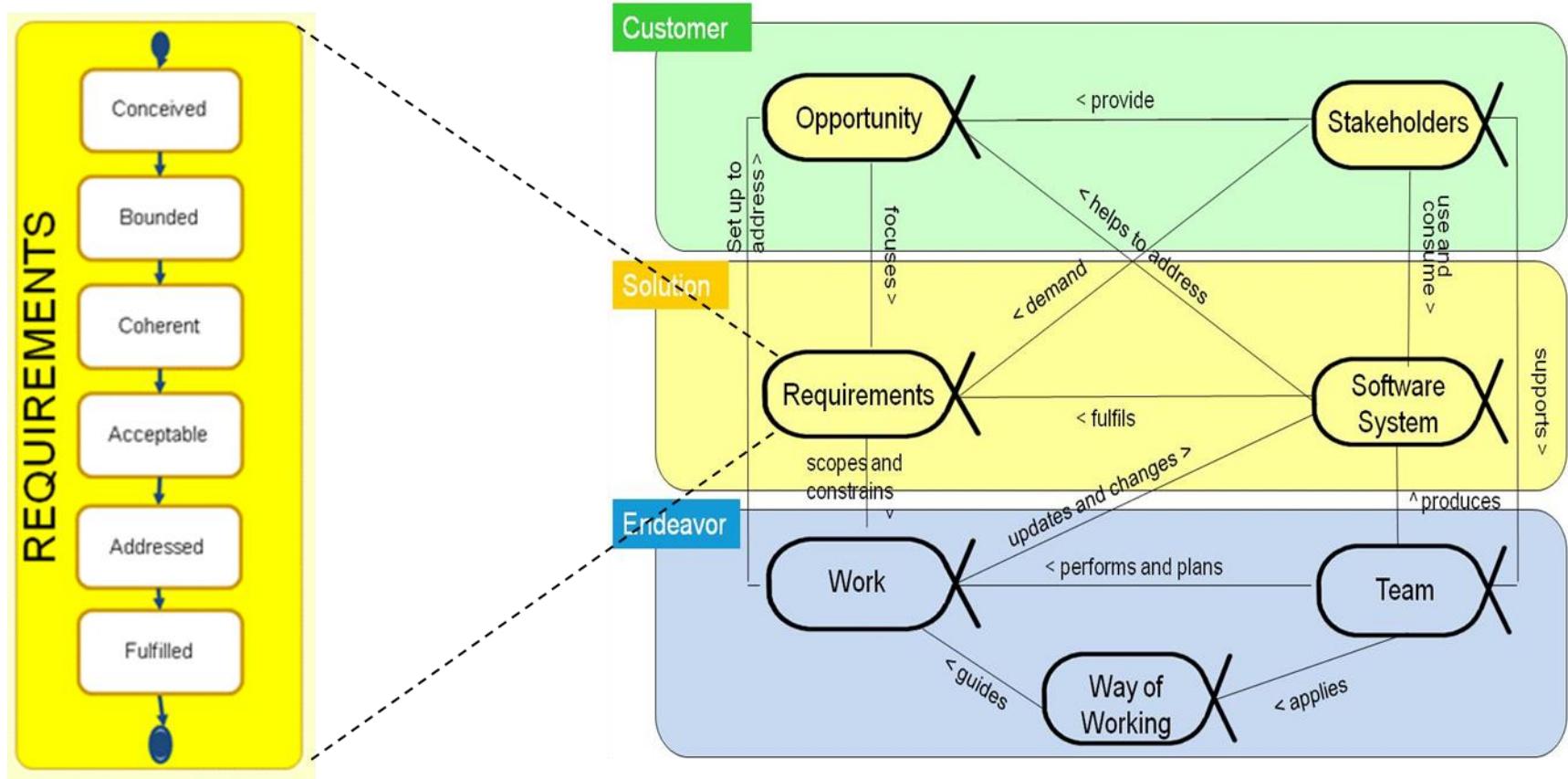
- Níveis
  - Auxilia
  - Aplica
  - Domina
  - Adapta
  - Inova



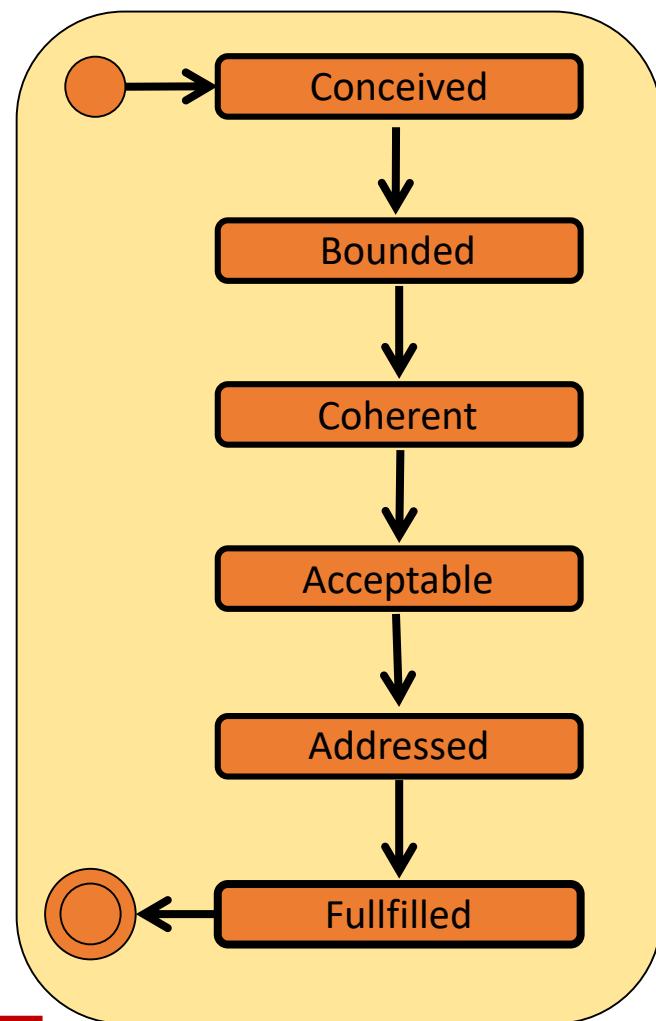
# Kernel: Competências



# Detalhamento em Estados



# Requirements – states



The need for a new system has been agreed.

The purpose and theme of the new system are clear.

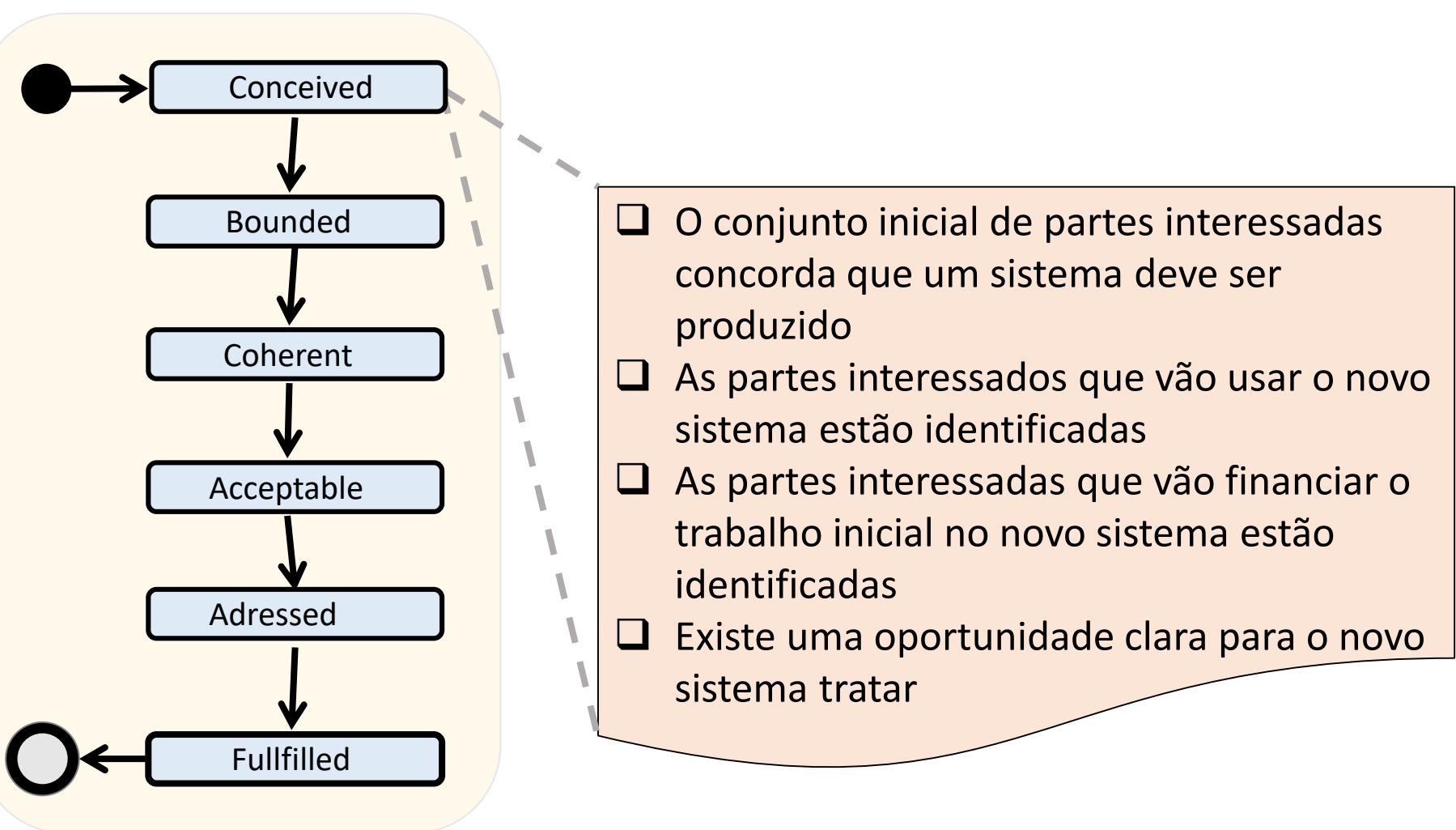
The requirements provide a coherent description of the essential characteristics of the new system.

The requirements describe a system that is acceptable to the stakeholders.

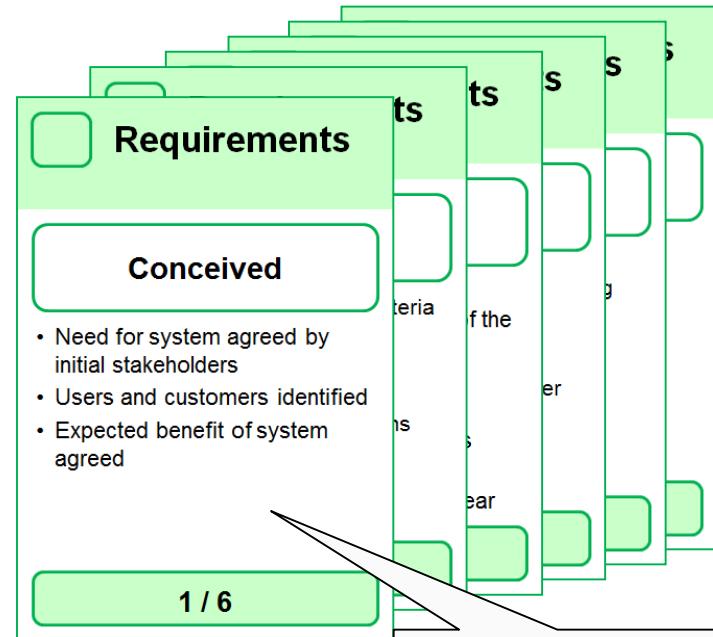
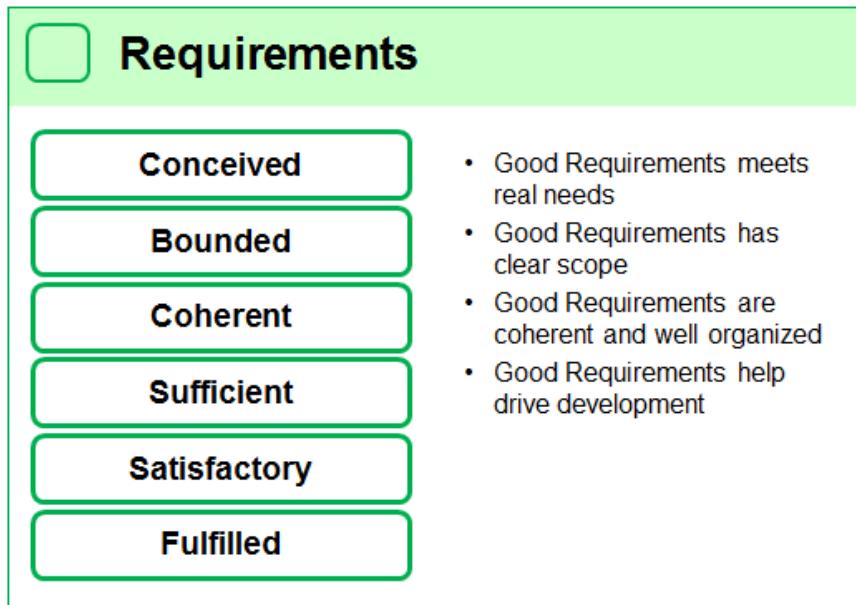
Enough of the requirements have been addressed to satisfy the need for a new system in a way that is acceptable to the stakeholders.

The requirements have been addressed to fully satisfy the need for a new system.

# Checklist para os estados de requisitos



# Orientado a resultado



# **Estados das Competências**

- Auxilia
- Aplica
- Domina
- Adapta
- Inova

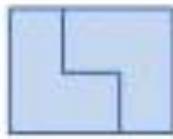
# **Espaços de Atividade**

- Representam coisa a fazer
- Complementam os Alphas para fornecer um visão baseada em atividades do desenvolvimento de software

# Evolução do Kernel

# Essence Lite

Checklists



aligns a set of >



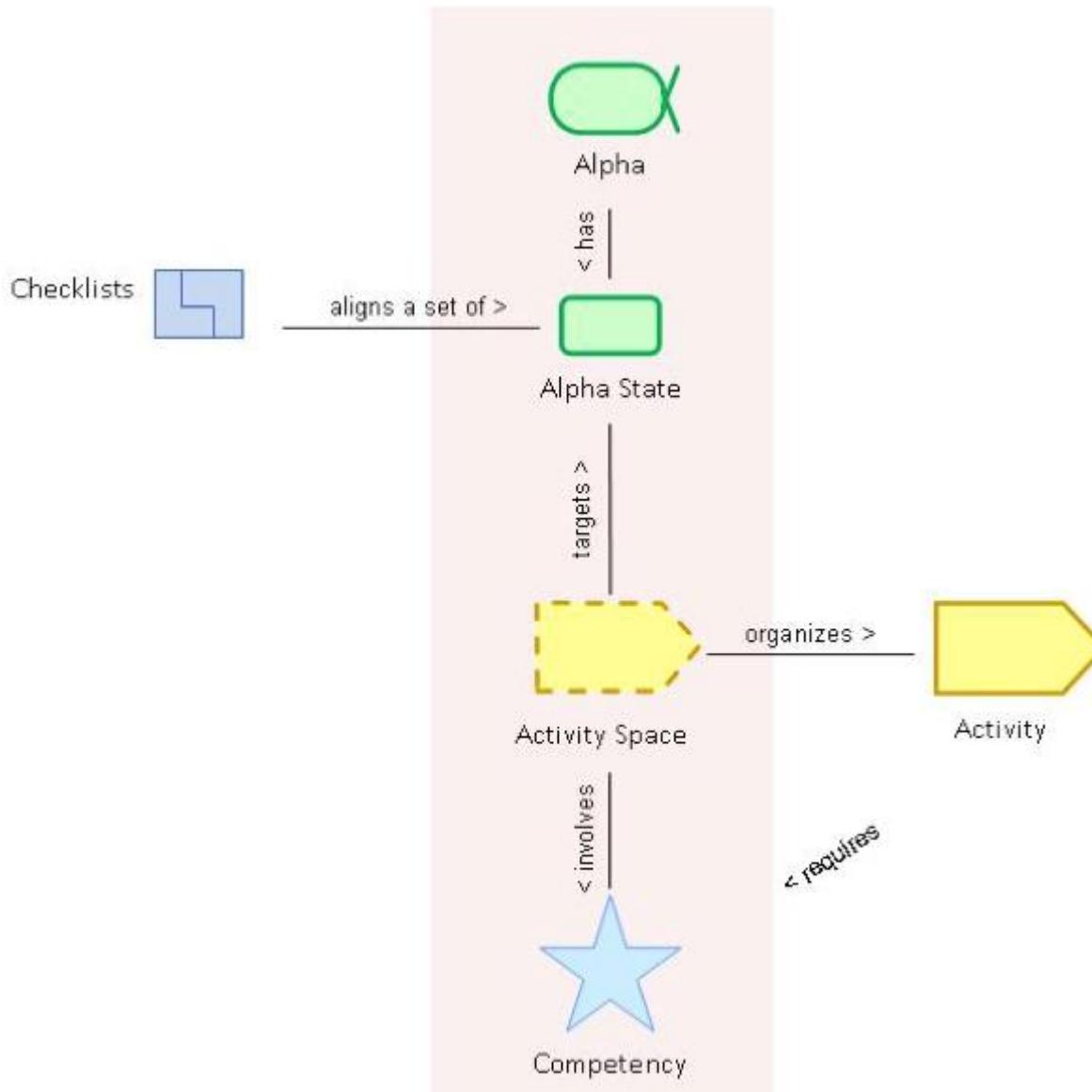
Alpha

has |

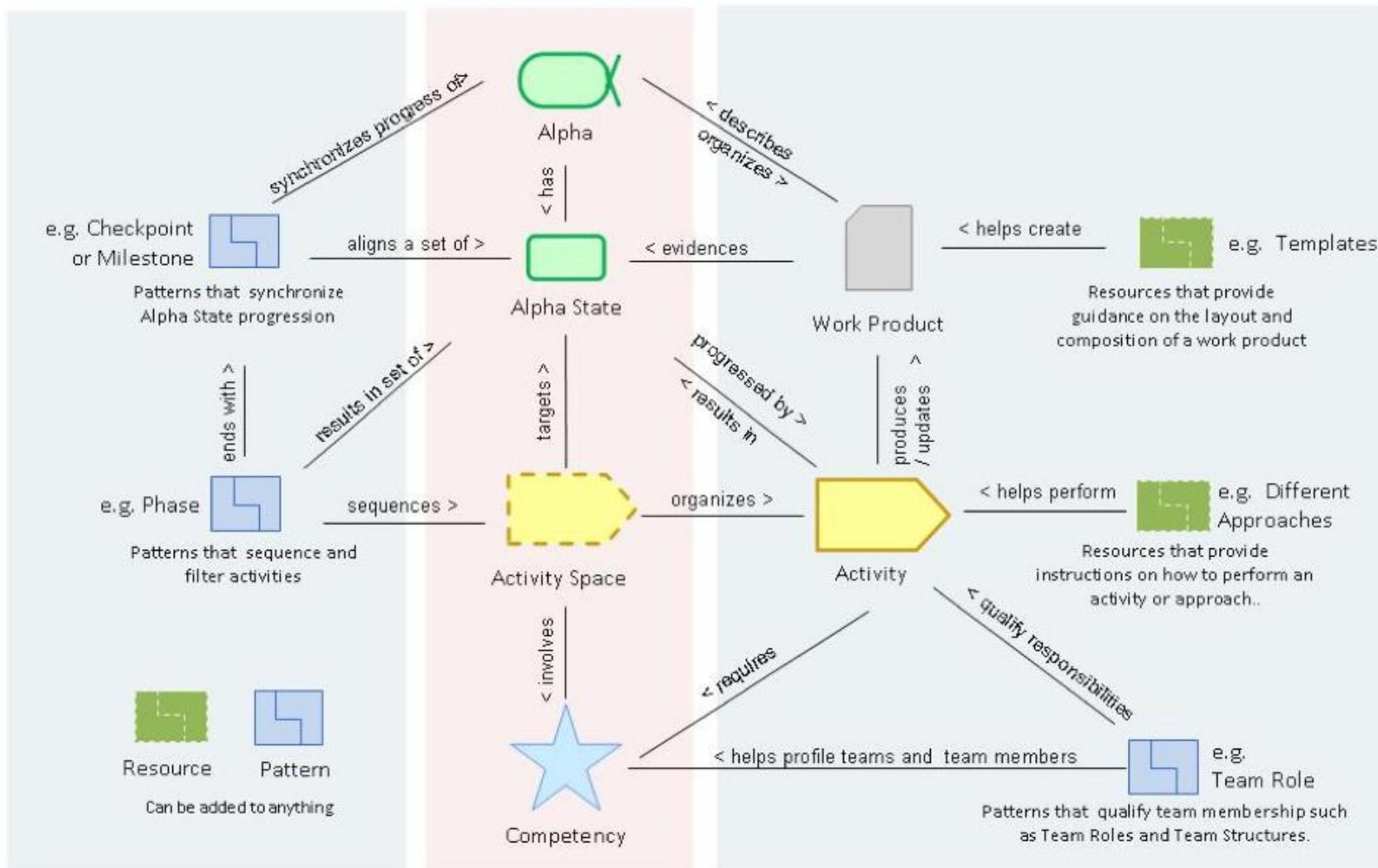


Alpha State

# Essence Kernel



# Essence Plus



# **Atividades: Cliente**

- Explorar Possibilidades
  - Inclui as oportunidades a serem abordadas e a identificação de partes interessadas
- Entender Necessidades das Partes Interessadas
  - EnGarantir a Satisfação das Partes Interessadas
- Usar o Sistema

# Cenários de Uso

# Compania X, Ana e Beto

- Company X executa vários projetos de desenvolvimento, grandes e pequenos
- Ana, a executiva, quer melhorar a capacidade de desenvolvimento
- Para isso, designa Beto
- Smith encontra com vários times, de diferentes backgrounds e experiências e os ajuda

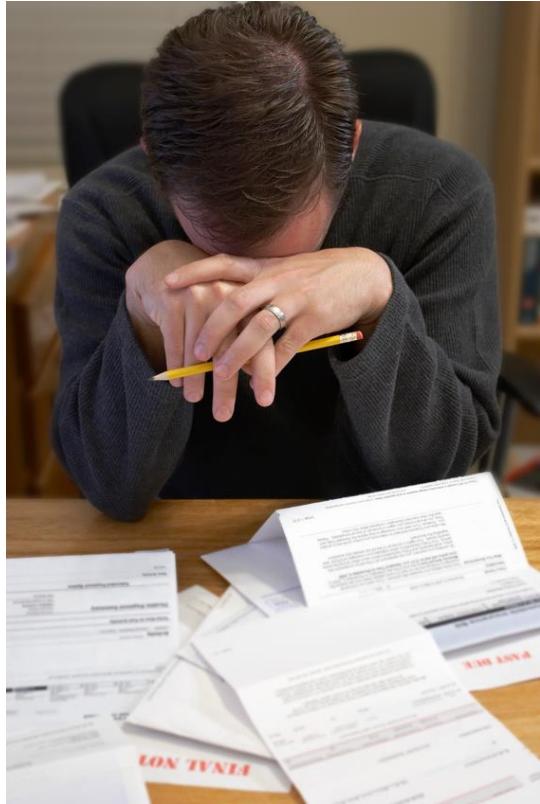


# Cenário: Encontro com time pequeno

- Beto encontro um time pequeno A
- Precisa entender o status do desenvolvimento
- Como fazer isso?
- Procurar por o que?
- Como chegar a uma conclusão?

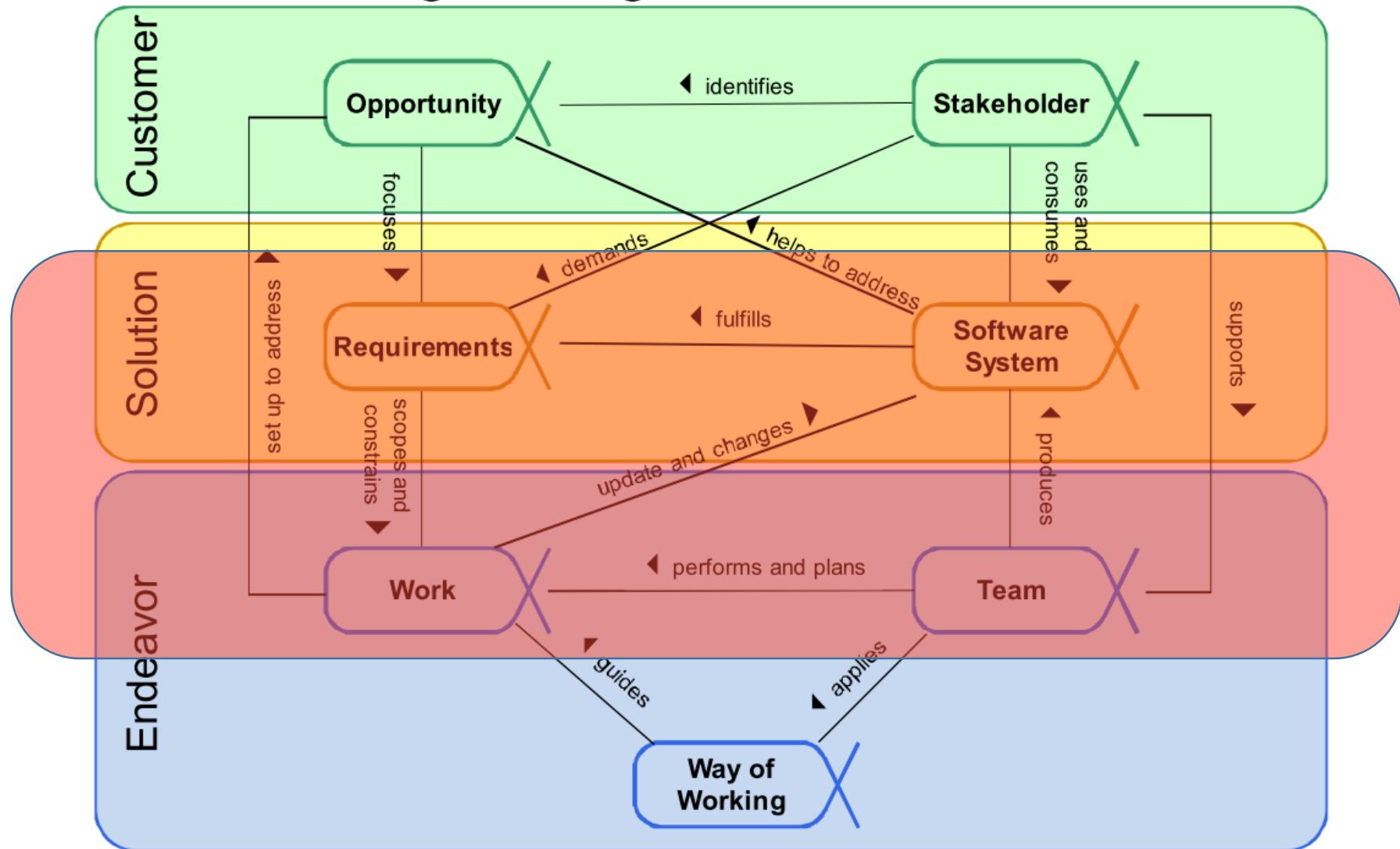


# Collecting the evidence



- O time dá documentos para Beto
  - Casos de Negócio
  - Modelo de Caso de Uso
  - Visão do Sistema
- Beto faz algumas perguntas
- O que perguntar?

# O Kernel - Alphas



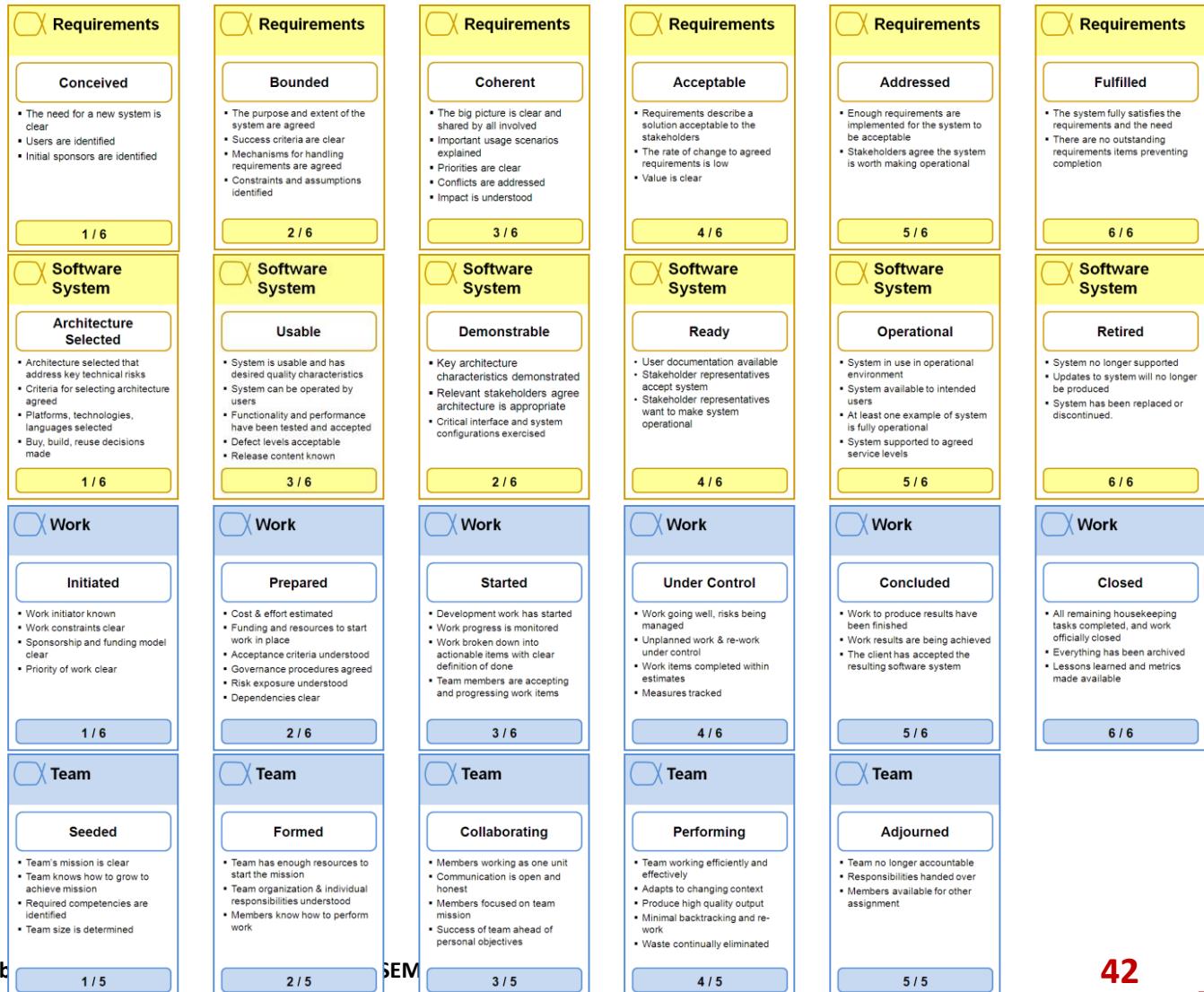
# Qual a situação atual?

Requirements

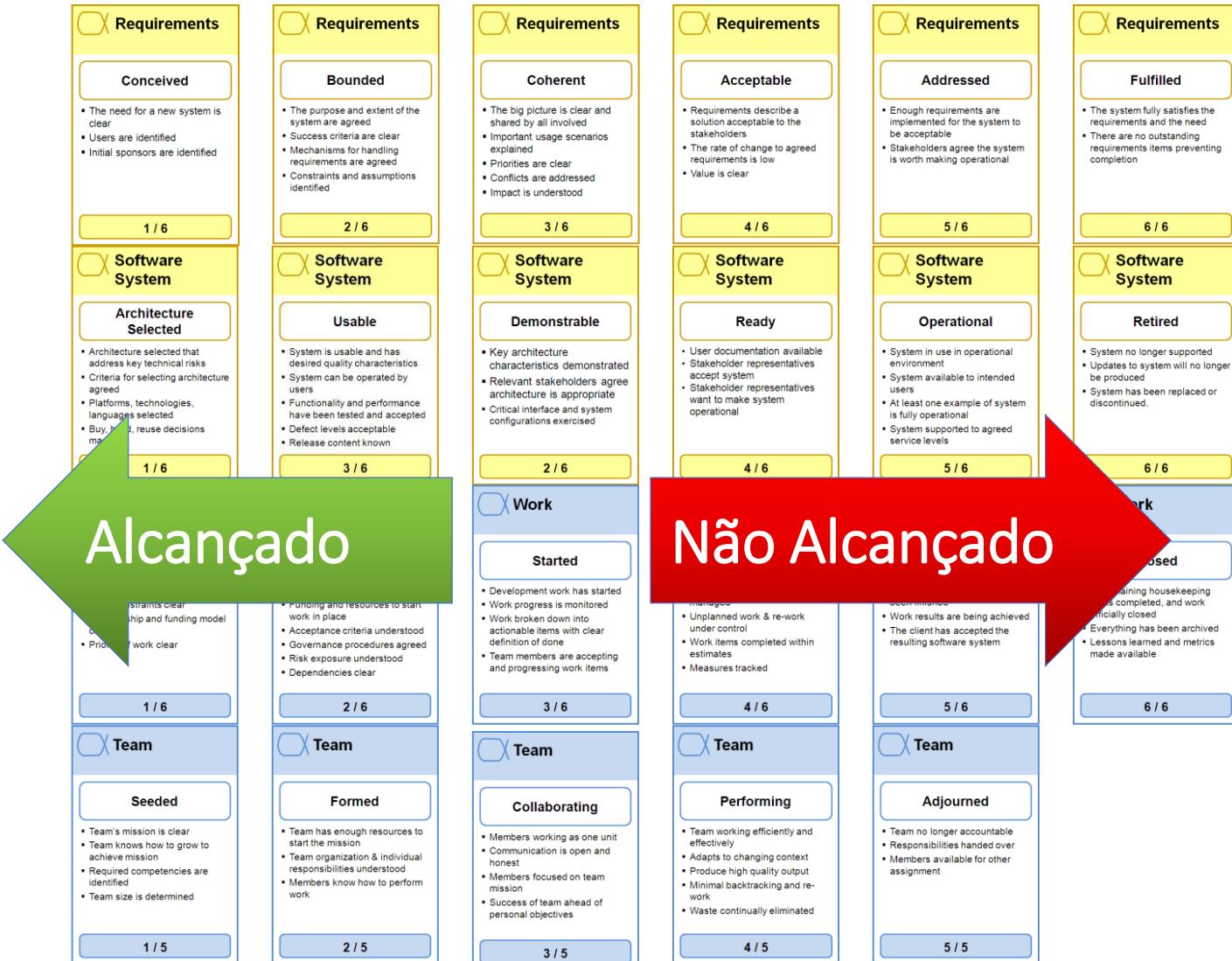
Software System

Work

Team



# Plano: Determine Estado Atual



# Estado Atual!

<p> Requirements</p> <p> Conceived</p> <ul style="list-style-type: none"> <li>The need for a new system is clear</li> <li>Users are identified</li> <li>Initial sponsors are identified</li> </ul> <p>1 / 6</p> <p> Software System</p> <p> Architecture Selected</p> <ul style="list-style-type: none"> <li>Architecture selected that address key technical risks</li> <li>Criteria for selecting architecture agreed</li> <li>Platforms, technologies, languages selected</li> <li>Buy, build, reuse decisions made</li> </ul> <p>1 / 6</p> <p> Work</p> <p> Initiated</p> <ul style="list-style-type: none"> <li>Work initiator known</li> <li>Work constraints clear</li> <li>Sponsorship and funding model clear</li> <li>Priority of work clear</li> </ul> <p>1 / 6</p> <p> Team</p> <p> Seeded</p> <ul style="list-style-type: none"> <li>Team's mission is clear</li> <li>Team knows how to grow to achieve mission</li> <li>Required competencies are identified</li> <li>Team size is determined</li> </ul> <p>1 / 5</p>	<p> Requirements</p> <p> Bounded</p> <ul style="list-style-type: none"> <li>The purpose and extent of the system are agreed</li> <li>Success criteria are clear</li> <li>Mechanisms for handling requirements are agreed</li> <li>Constraints and assumptions identified</li> </ul> <p>2 / 6</p> <p> Software System</p> <p> Usable</p> <ul style="list-style-type: none"> <li>System is usable and has desired quality characteristics</li> <li>System can be operated by users</li> <li>Functionality and performance have been tested and accepted</li> <li>Defect levels acceptable</li> <li>Release content known</li> </ul> <p>3 / 6</p> <p> Work</p> <p> Prepared</p> <ul style="list-style-type: none"> <li>Cost &amp; effort estimated</li> <li>Funding and resources to start work in place</li> <li>Acceptance criteria understood</li> <li>Governance procedures agreed</li> <li>Risk exposure understood</li> <li>Dependencies clear</li> </ul> <p>2 / 6</p> <p> Team</p> <p> Formed</p> <ul style="list-style-type: none"> <li>Team has enough resources to start the mission</li> <li>Team organization &amp; individual responsibilities understood</li> <li>Members know how to perform work</li> </ul> <p>2 / 5</p>	<p> Requirements</p> <p> Coherent</p> <ul style="list-style-type: none"> <li>The big picture is clear and shared by all involved</li> <li>Important usage scenarios explained</li> <li>Priorities are clear</li> <li>Conflicts are addressed</li> <li>Impact is understood</li> </ul> <p>3 / 6</p> <p> Software System</p> <p> Demonstrable</p> <ul style="list-style-type: none"> <li>Key architecture characteristics demonstrated</li> <li>Relevant stakeholders agree architecture is appropriate</li> <li>Critical interface and system configurations exercised</li> </ul> <p>2 / 6</p> <p> Work</p> <p> Started</p> <ul style="list-style-type: none"> <li>Development work has started</li> <li>Work progress is monitored</li> <li>Work broken down into actionable items with clear definition of done</li> <li>Team members are accepting and progressing work items</li> </ul> <p>3 / 6</p> <p> Team</p> <p> Collaborating</p> <ul style="list-style-type: none"> <li>Members working as one unit</li> <li>Communication is open and effective</li> <li>Members focused on team mission</li> <li>Success of team ahead of personal objectives</li> </ul> <p>3 / 5</p>	<p> Requirements</p> <p> Acceptable</p> <ul style="list-style-type: none"> <li>Requirements describe a solution acceptable to the stakeholders</li> <li>The rate of change to agreed requirements is low</li> <li>Value is clear</li> </ul> <p>4 / 6</p> <p> Software System</p> <p> Ready</p> <ul style="list-style-type: none"> <li>User documentation available</li> <li>Stakeholder representatives accept system</li> <li>Stakeholder representatives want to make system operational</li> </ul> <p>4 / 6</p> <p> Work</p> <p> Under Control</p> <ul style="list-style-type: none"> <li>Work going well, risks being managed</li> <li>Unplanned work &amp; re-work under control</li> <li>Work items completed within estimates</li> <li>Measures tracked</li> </ul> <p>4 / 6</p> <p> Team</p> <p> Performing</p> <ul style="list-style-type: none"> <li>Team working efficiently and effectively</li> <li>Adapts to changing context</li> <li>Produce high quality output</li> <li>Minimal backtracking and re-work</li> <li>Waste continually eliminated</li> </ul> <p>4 / 5</p>	<p> Requirements</p> <p> Addressed</p> <ul style="list-style-type: none"> <li>Enough requirements are implemented for the system to be acceptable</li> <li>Stakeholders agree the system is worth making operational</li> </ul> <p>5 / 6</p> <p> Software System</p> <p> Operational</p> <ul style="list-style-type: none"> <li>System in use in operational environment</li> <li>System available to intended users</li> <li>At least one example of system is fully operational</li> <li>System supported to agreed service levels</li> </ul> <p>5 / 6</p> <p> Work</p> <p> Concluded</p> <ul style="list-style-type: none"> <li>Work to produce results have been finished</li> <li>Work results are being achieved</li> <li>The client has accepted the resulting software system</li> </ul> <p>5 / 6</p> <p> Team</p> <p> Adjourned</p> <ul style="list-style-type: none"> <li>Team no longer accountable</li> <li>Responsibilities handed over</li> <li>Members available for other assignment</li> </ul> <p>5 / 5</p>	<p> Requirements</p> <p> Fulfilled</p> <ul style="list-style-type: none"> <li>The system fully satisfies the requirements and the need</li> <li>There are no outstanding requirements items preventing completion</li> </ul> <p>6 / 6</p> <p> Software System</p> <p> Retired</p> <ul style="list-style-type: none"> <li>System no longer supported</li> <li>Updates to system will no longer be produced</li> <li>System has been replaced or discontinued.</li> </ul> <p>6 / 6</p> <p> Work</p> <p> Closed</p> <ul style="list-style-type: none"> <li>All remaining housekeeping tasks completed, and work officially closed</li> <li>Everything has been archived</li> <li>Lessons learned and metrics made available</li> </ul> <p>6 / 6</p>
---	---	---	---	--	--

# Cliente

- Todo projeto de desenvolvimento de software tem ao menos um cliente para o software que ele produz
- Qualquer processo adotado precisa integrar a perspectiva do cliente no projeto para prevenir que uma solução não apropriada seja produzida
- Lembrem do CHAOS Report:

**FIM**