

SPLICE- A FLEXIBLE SPL LIFECYCLE MANAGEMENT TOOL

APRESENTADO POR BRUNO CABRAL

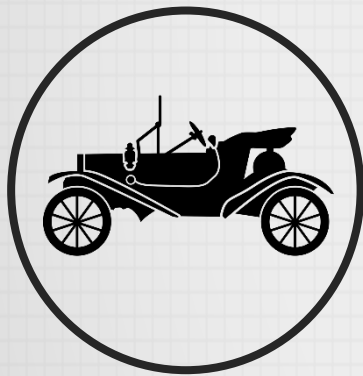
ORIENTADOR: EDUARDO SANTANA DE ALMEIDA

CO-ORIENTADOR: TASSIO VALE

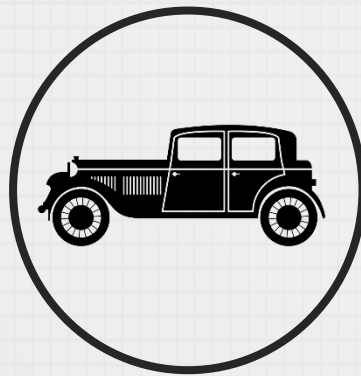


MOTIVAÇÃO LINHA DE PRODUTOS DE SOFTWARE

- Software está tornando cada vez mais complexo
- **Reuso** começa a ser obrigatório
- **Customização em massa** – bens e serviços para atender às necessidades individuais do cliente – e que devem ser feitas com eficiência de produção em massa



Feitos a mão



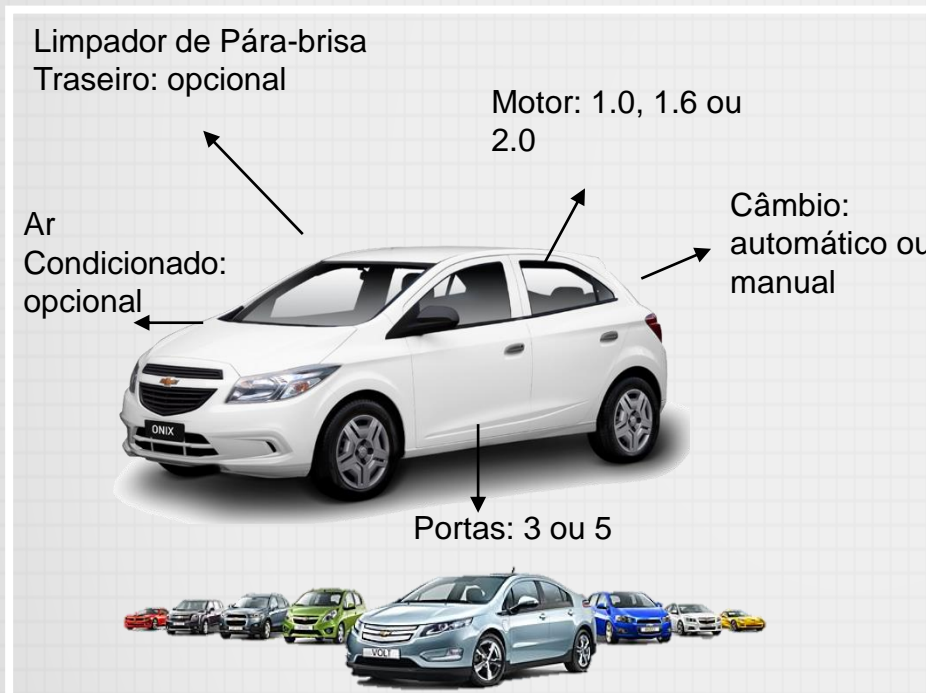
Produzidos
em massa



Customizados
em massa

EXPERIÊNCIA DA INDÚSTRIA

Muitas organizações produzem famílias de sistemas similares, diferenciados pelo conjunto de recursos. Um exemplo recorrente é a linha de produtos de carros.



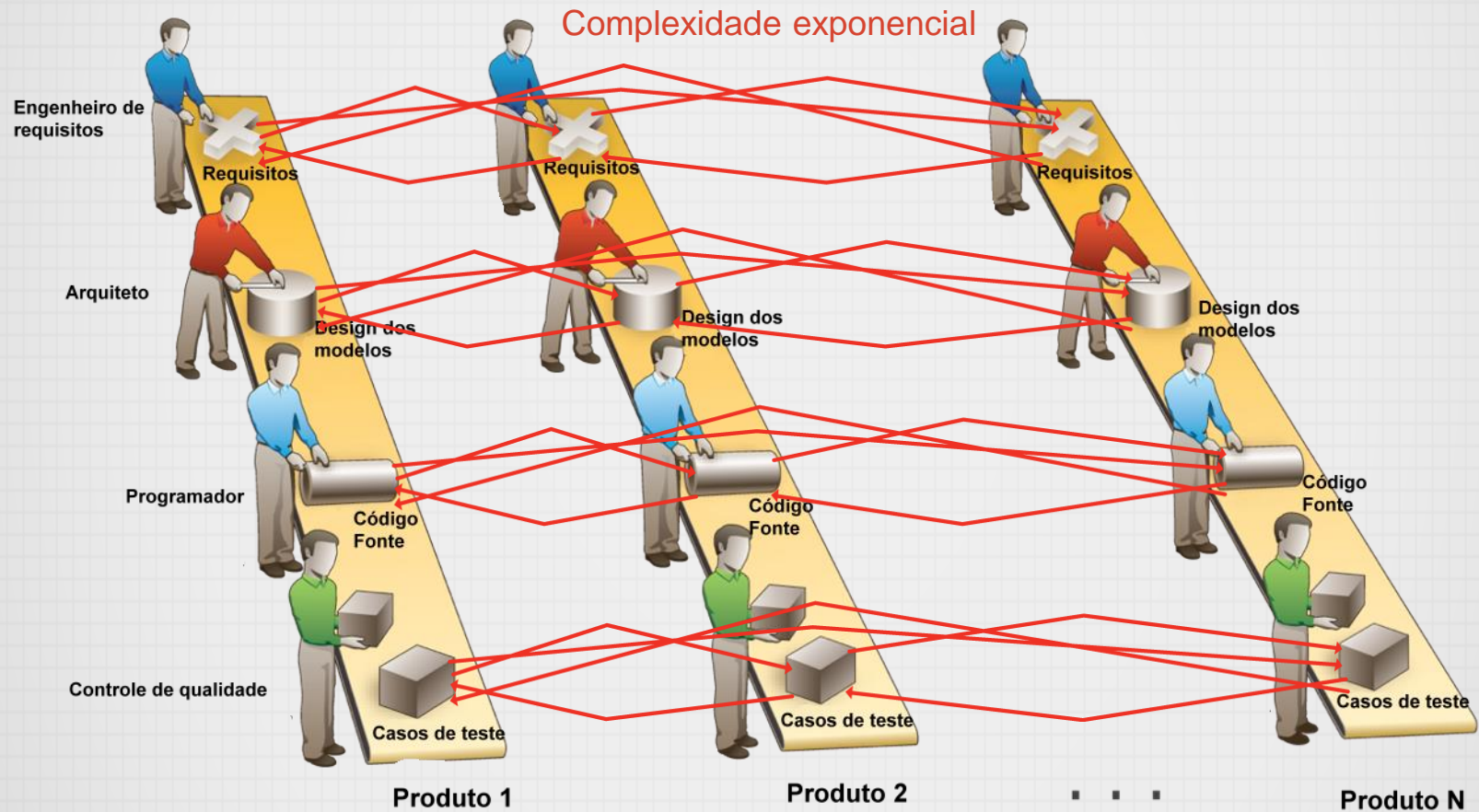
EMPRESA

General Motors

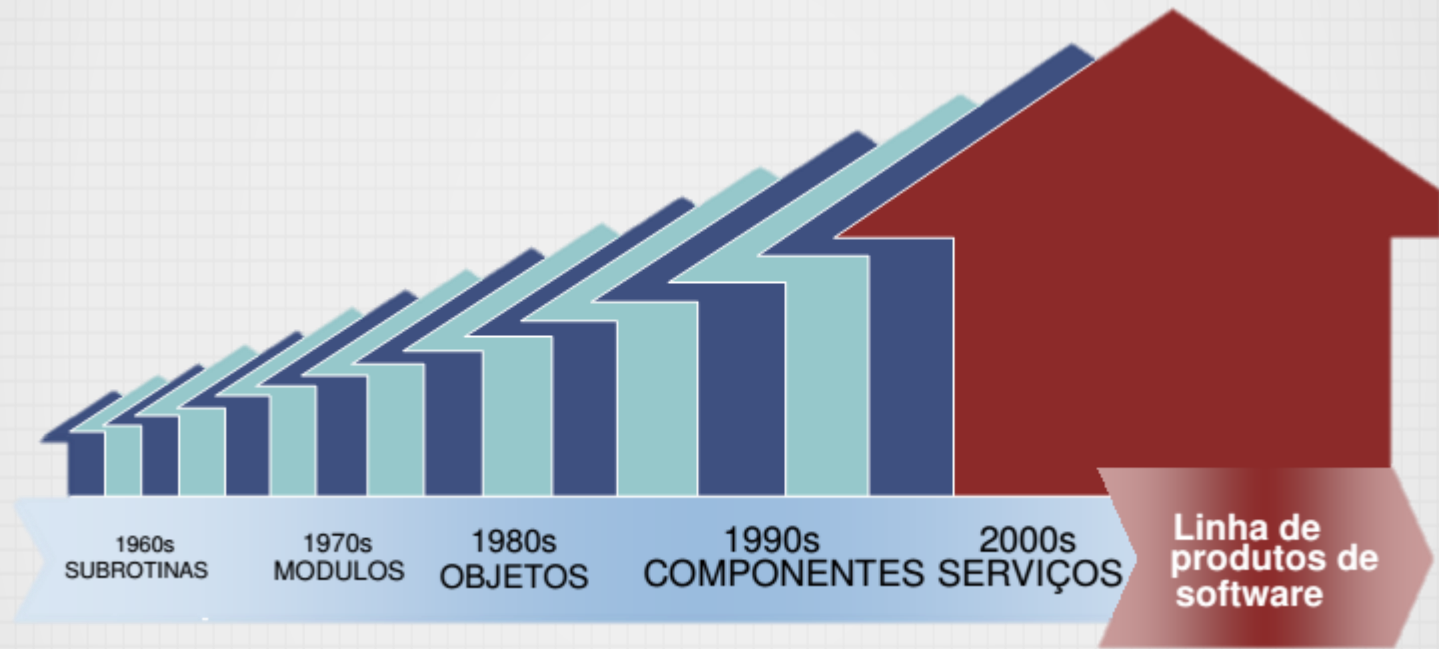
SITUAÇÃO

- A GM é a maior empresa de automóveis do mundo.
- Um veículo da GM é composto por mais de 300 subsistemas como freios, iluminação, etc...
- O Chevrolet Volt possui mais de 10 milhões de linhas de código, mais que um Boeing 787.
- A GM produz mais de 60 modelos de automóveis, sem contar a legislação e preferências culturais dos 150 países que atua.

PROCESSO NA GM



LINHA DO TEMPO DO REUSO



DEFINIÇÃO LINHA DE PRODUTOS DE SOFTWARE

Linha de produtos de software é um **conjunto** de sistemas com uso intensivo de software que compartilham um conjunto comum de características gerenciáveis para satisfazer necessidades específicas de um particular segmento de mercado, ou missão, e que são desenvolvidos a partir de um conjunto de core assets de uma maneira prescrita.

LINDA

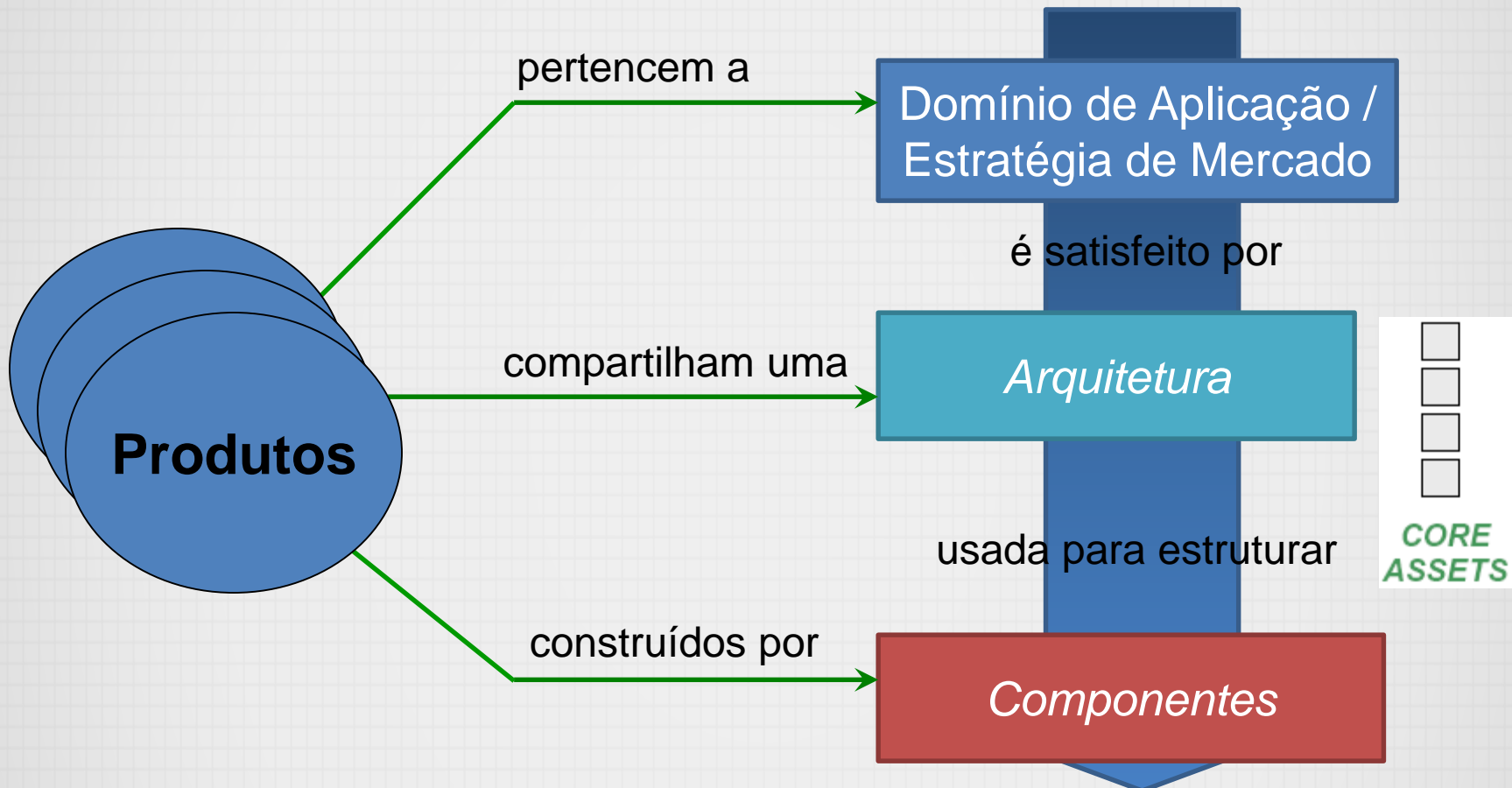
Engenharia de linha de produtos de software é um paradigma para o desenvolvimento de aplicações de software (sistemas com uso intensivo de software e produtos de software), utilizando plataformas e customização em massa.

POHL

BENEFÍCIOS

- Redução dos Custos de Desenvolvimento
- Aumento da Qualidade
- Redução do time-to-market
- Ampliação do portfolio

LINHA DE PRODUTOS DE SOFTWARE

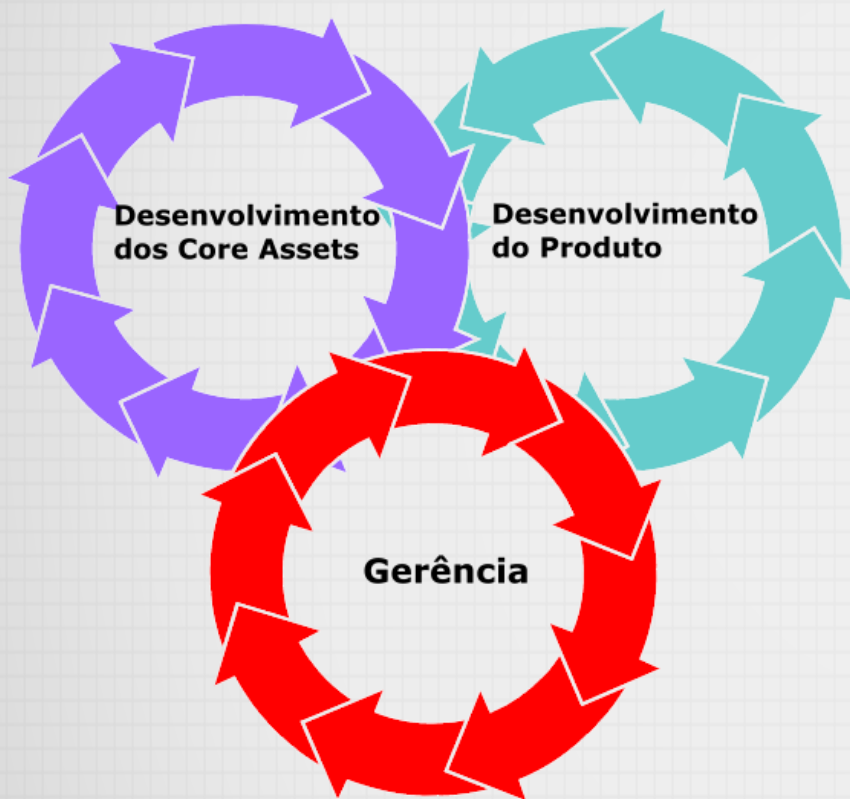


Linhas de Produtos:

Tiram vantagens econômicas sobre partes comuns (*commonality*)

Ligam (*bound*) a variabilidade

ATIVIDADES ESSENCIAIS



INTERATIVAS

As atividades são interativas e suas saídas e entradas influenciam umas as outras.

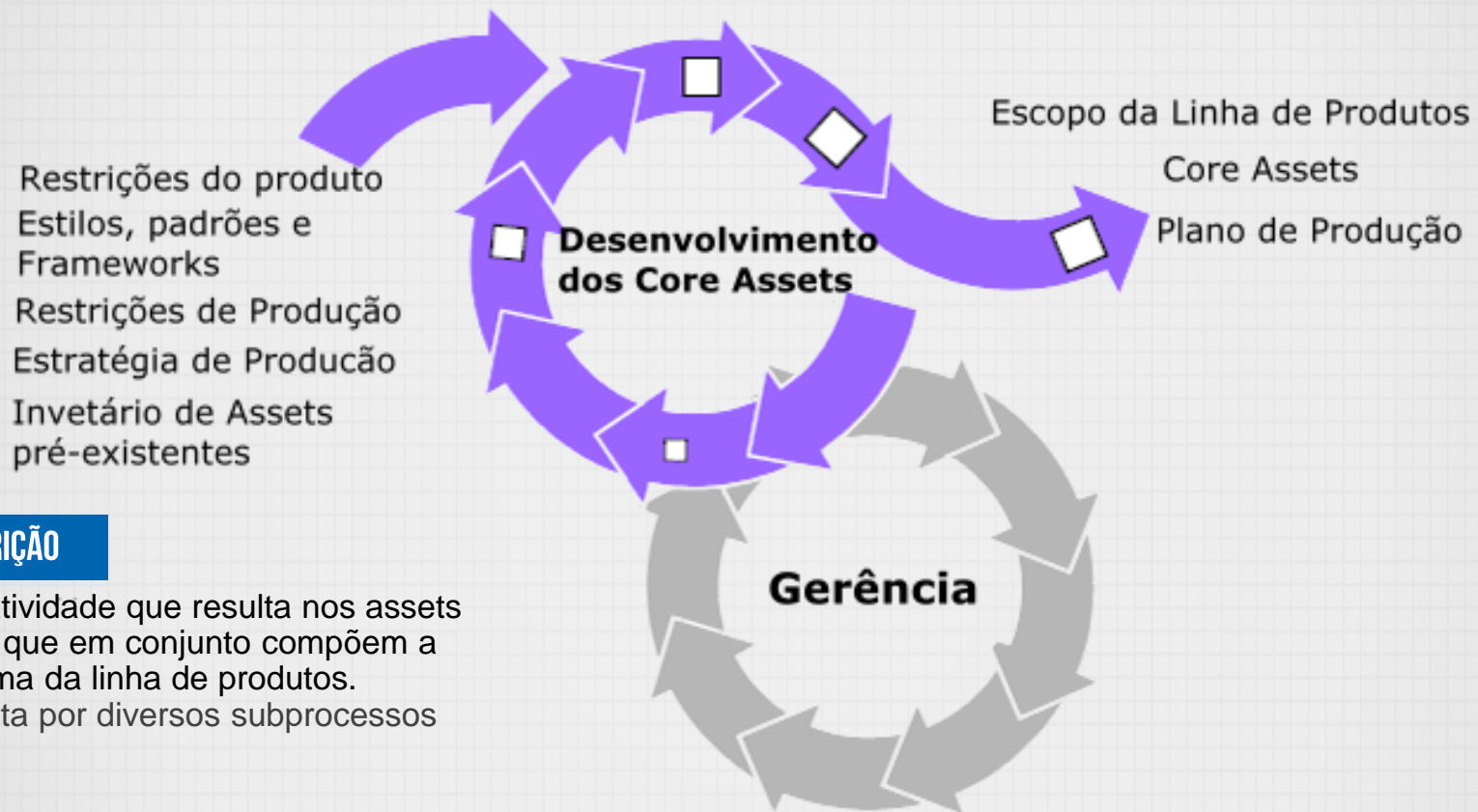
FEEDBACK CONTINUO

Caso os artefatos desenvolvidos em uma etapa anterior não estejam adequados, precisam ser refeitos. O feedback contínuo é extremamente importante para manter a qualidade.

GERENCIAMENTO

É necessário um gerenciamento contínuo e rigoroso durante as etapas. Isso é determinante para o sucesso da LPS. Inclui fatores organizacionais e técnicos.

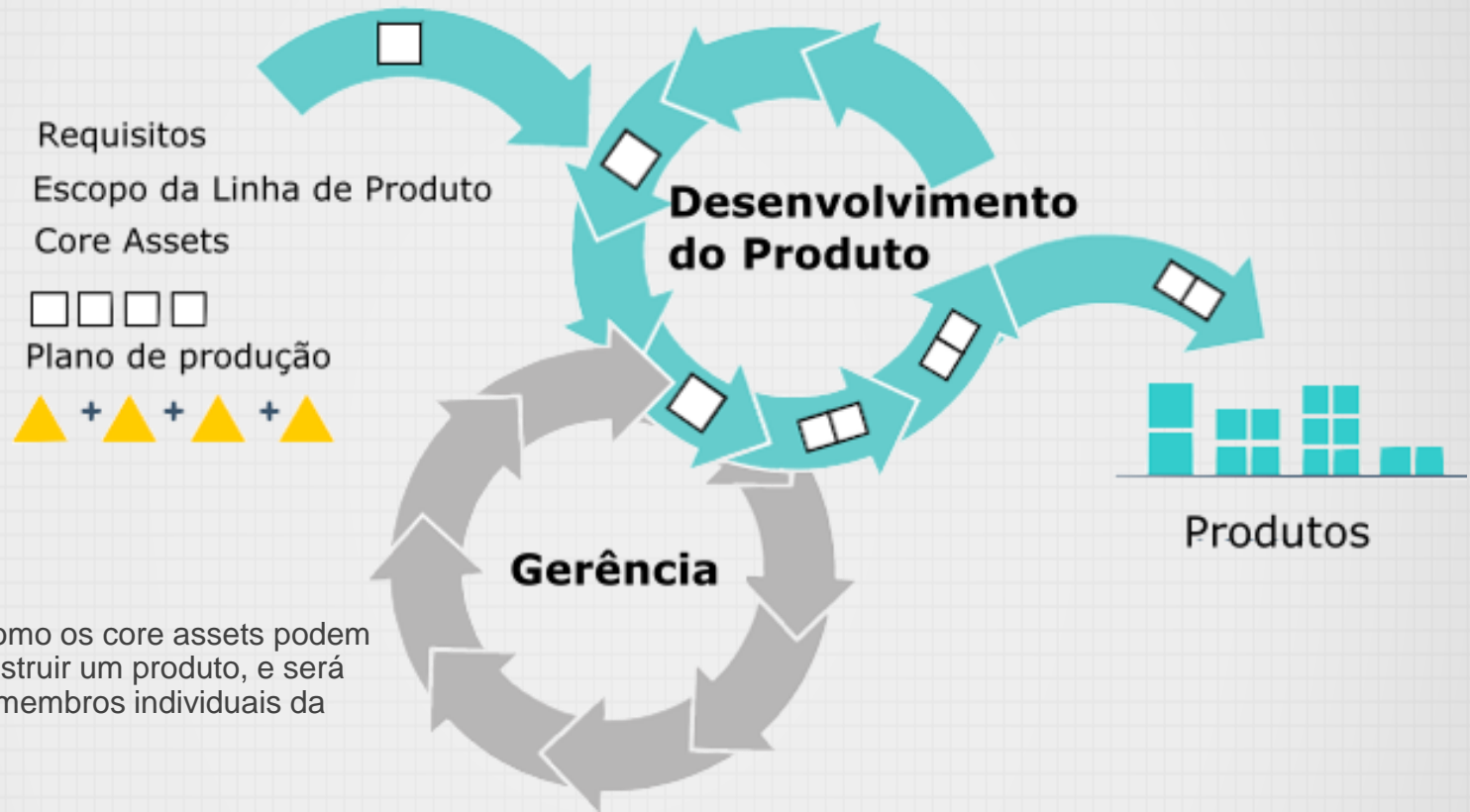
DESENVOLVIMENTO DOS CORE ASSETS



DESCRIÇÃO

É uma atividade que resulta nos assets comuns que em conjunto compõem a plataforma da linha de produtos.
Composta por diversos subprocessos

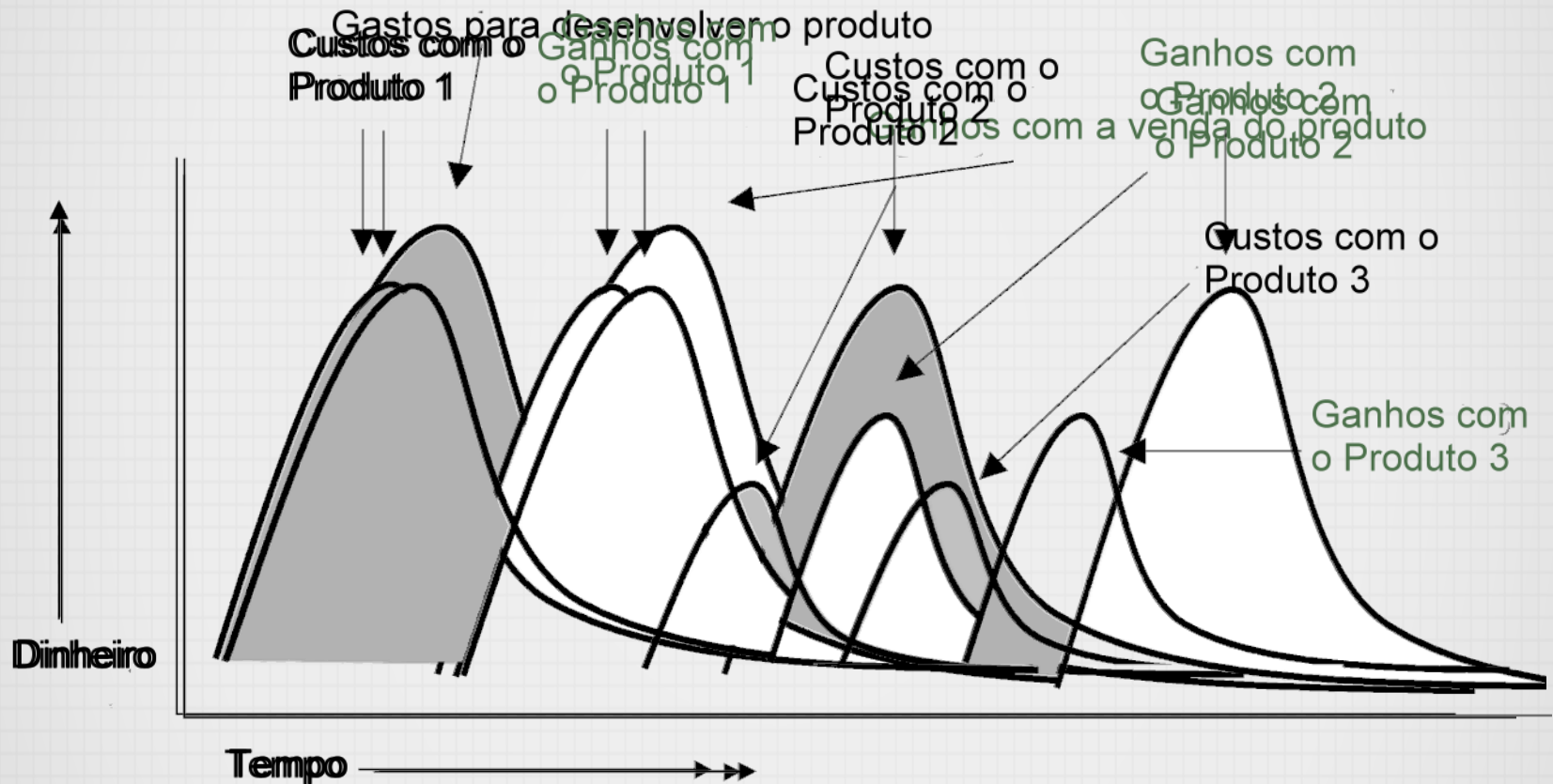
DESENVOLVIMENTO DO PRODUTO



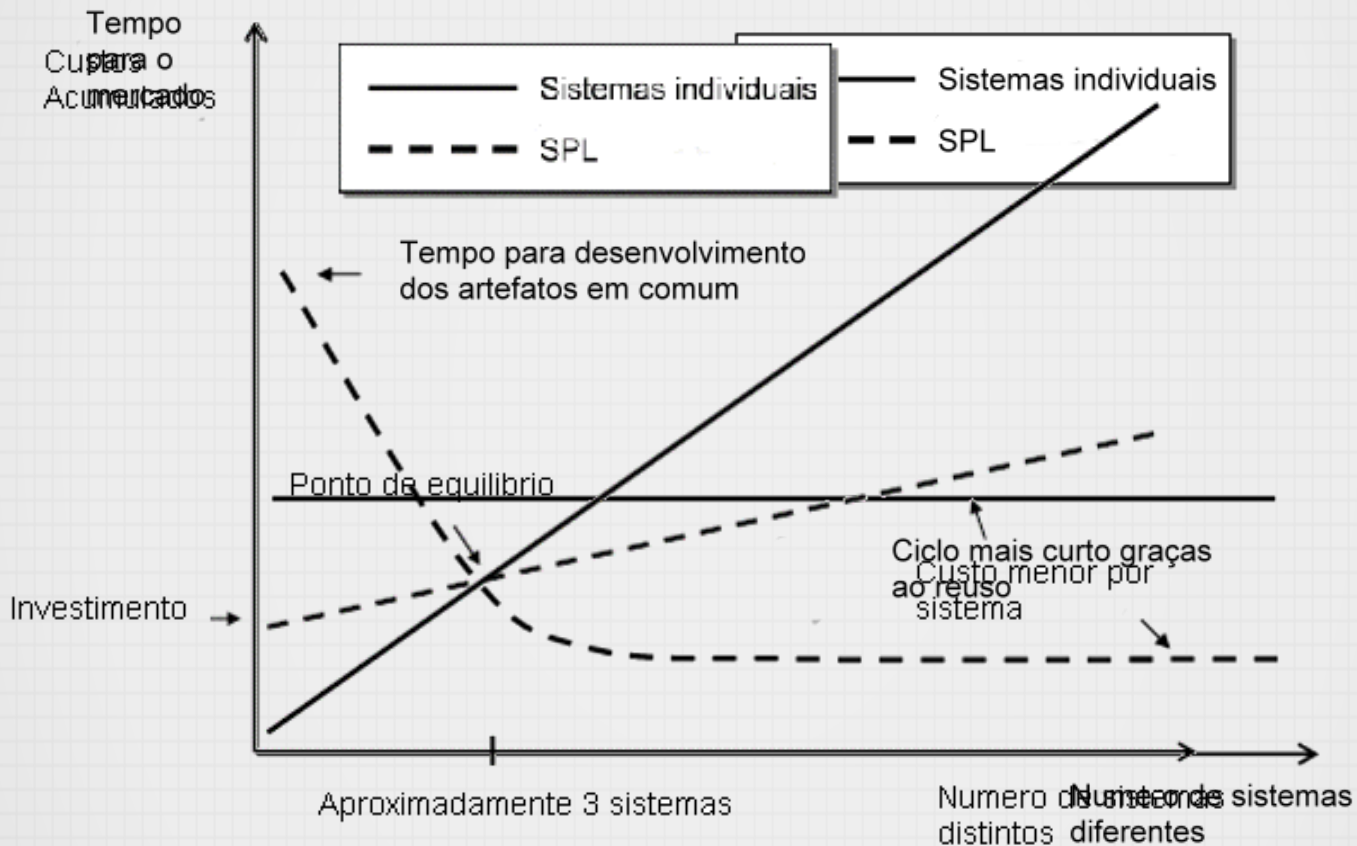
DESCRIÇÃO

Essa fase detalha como os core assets podem ser usados para construir um produto, e será usado para criar os membros individuais da linha de produtos.

TEMPO X DINHEIRO

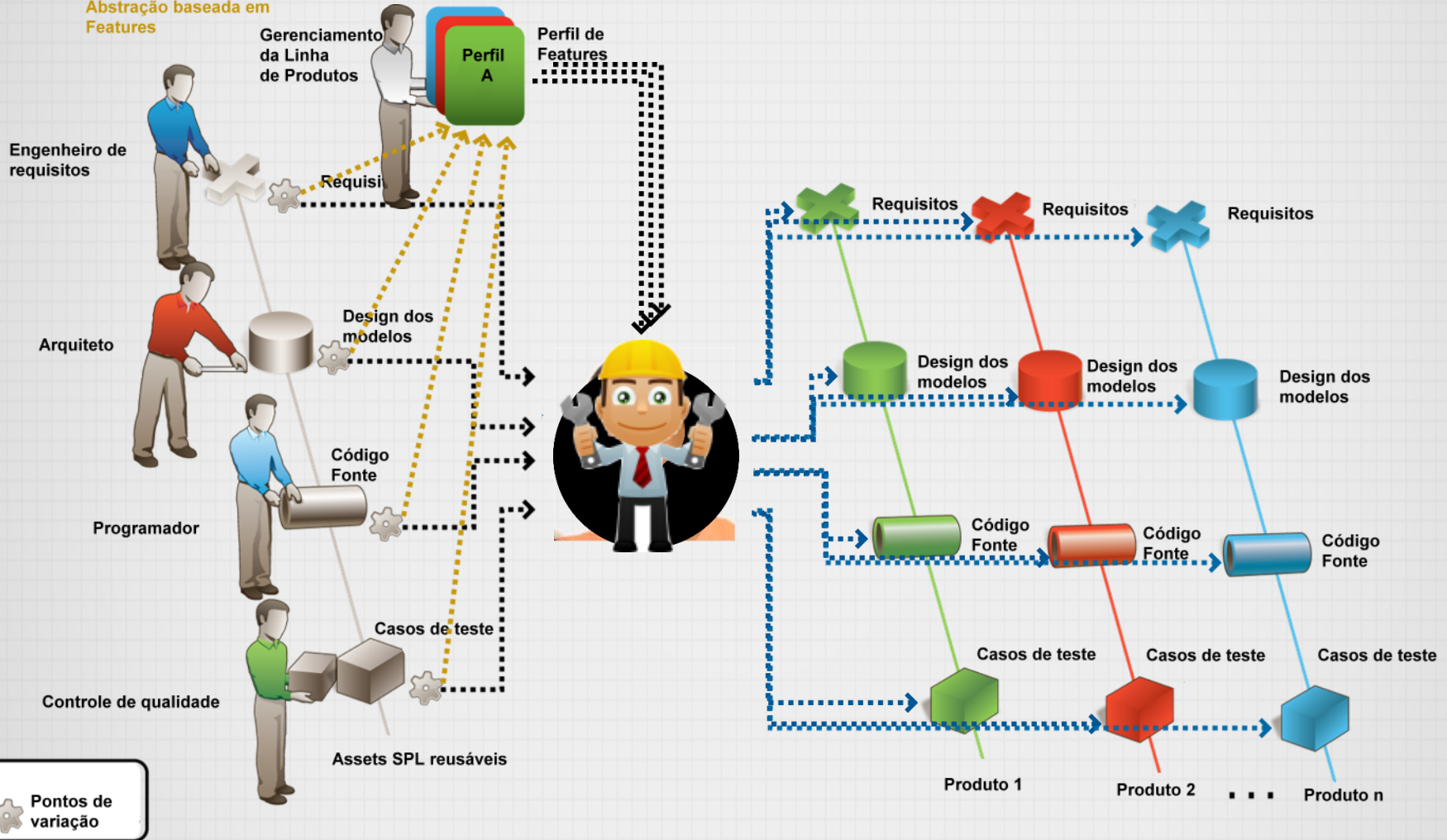


TEMPO X DINHEIRO 2

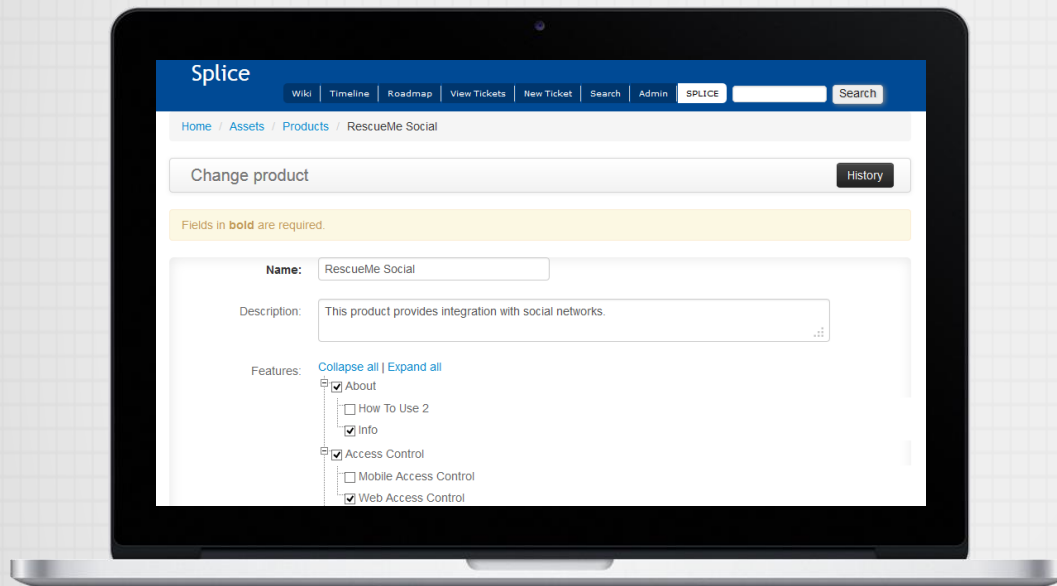


EVOLUÇÃO DO PROCESSO NA GM

Abstração baseada em Features



CONHEÇAM O SPLICE

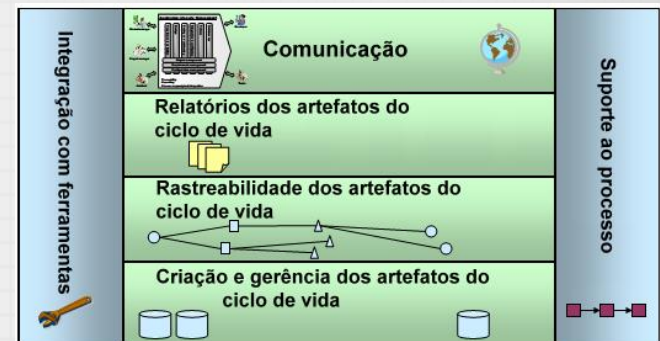


AMBIENTE CONSTRUÇÃO INTEGRADO DE LINHA DE PRODUTO DE SOFTWARE (SPLICE)

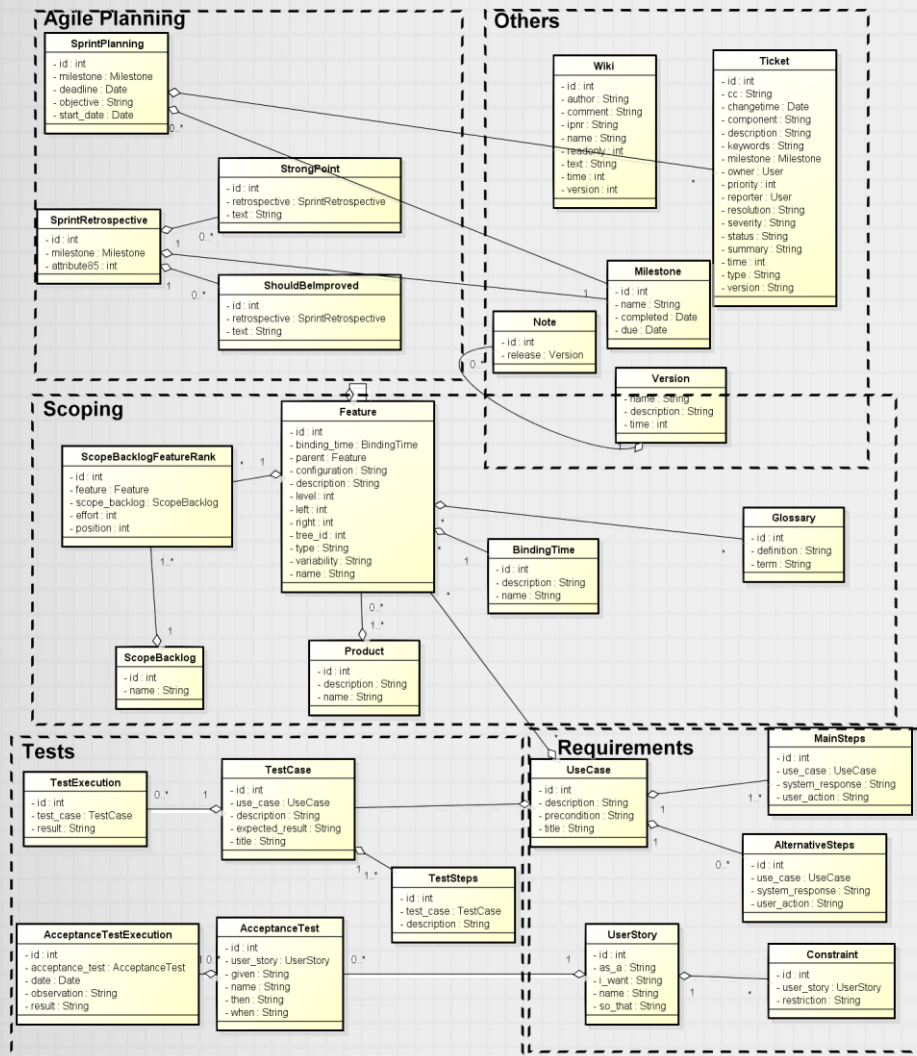
DESCRIÇÃO

É um sistema de gerenciamento de ciclo de vida de aplicativos (ALM) feito para apoiar e integrar as principais atividades de LPS, como, gerenciamento de requisitos, arquitetura, codificação, testes, controle e gerenciamento de versões, proporcionando a automação de atividades e **rastreabilidade** em todo o processo.

FRAMEWORK ALM USADO

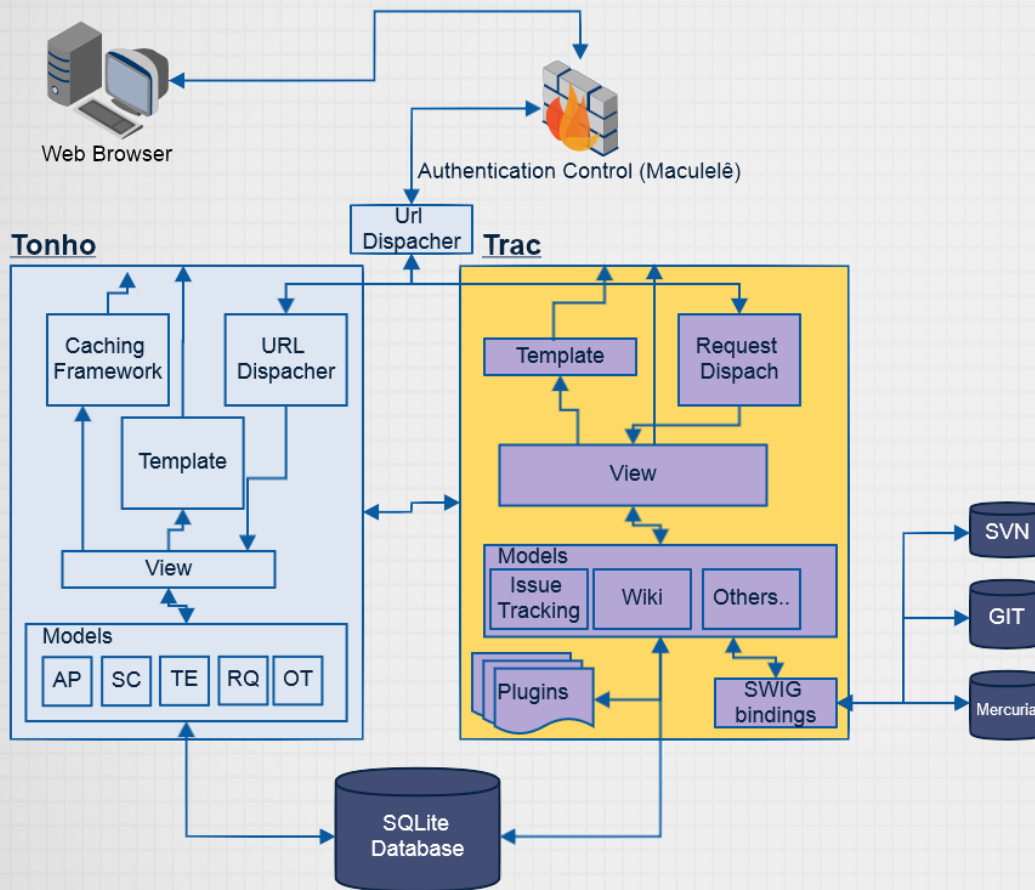


O METAMODELO LEVE



- Usamos uma abordagem baseada em modelos para representar todas as informações, atividades e conexões entre os artefatos.
- Nenhum metamodelo proposto anteriormente integrava métodos ágeis.
- Foi proposto nesse trabalho o “Metamodelo leve de LPS”, que foi implementado na SPLICE

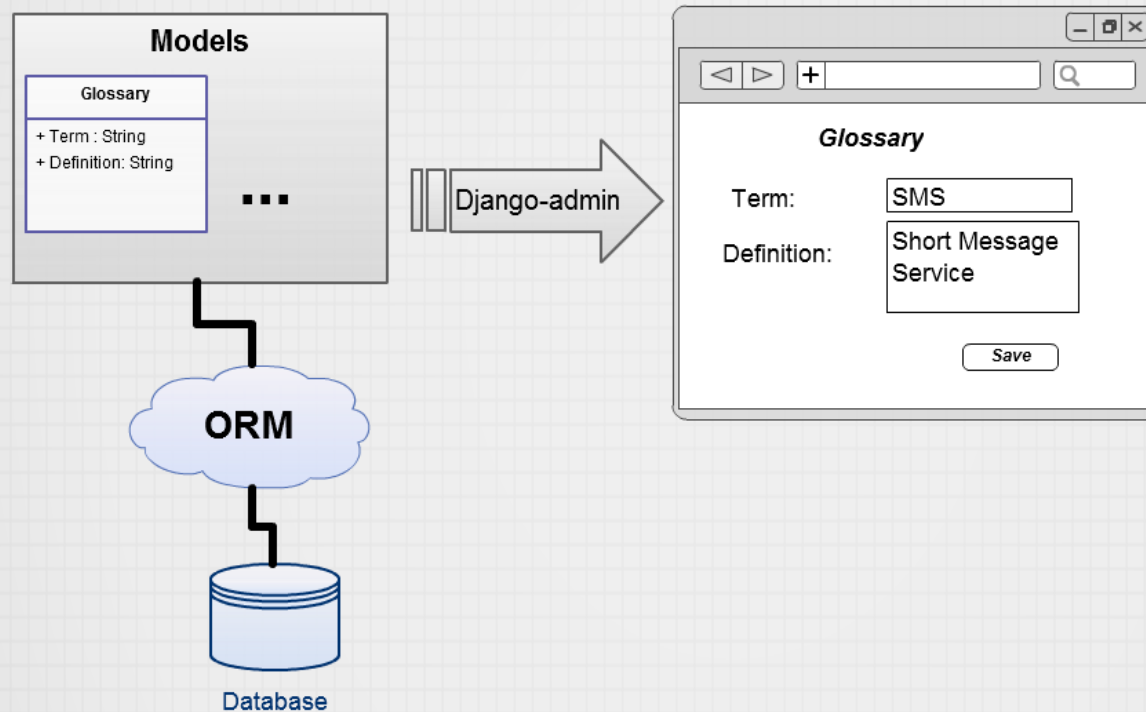
A ARQUITETURA



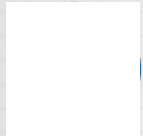
- Composto de 3 módulos
 - Maculelê
 - Tonho
 - Trac
- Todos escritos em Python. E com exceção do “Trac”, todos utilizam o framework Django.
- “Trac” foi estendido para acomodar nossas mudanças através de plug-ins.
- Meta-modelo implementado usando ORM e geradores de aplicação.

TRANSFORMAÇÃO DO MODELO

17



REQUISITOS NÃO-FUNCIONAIS



- **INTERFACE WEB**
- Permite que engenheiros e stakeholders colaborem e acessem o sistema de qualquer localização.



- **EXTENSIBILIDADE**
- Deve permitir que o adicione funcionalidades através de uma interface publica.



- **USABILIDADE**
- Deve oferecer uma experiência para o usuário consistente.



- **RESPONSABILIDADE**
- Todas as ações devem ser registradas para atribuição de responsabilidade.



- **METAMODELO FLEXÍVEL**
- Deve permitir que se altere facilmente o metamodelo



- **TRANSPARÊNCIA**
- O usuário deve visualizar o conjunto de ferramentas integradas como uma só.

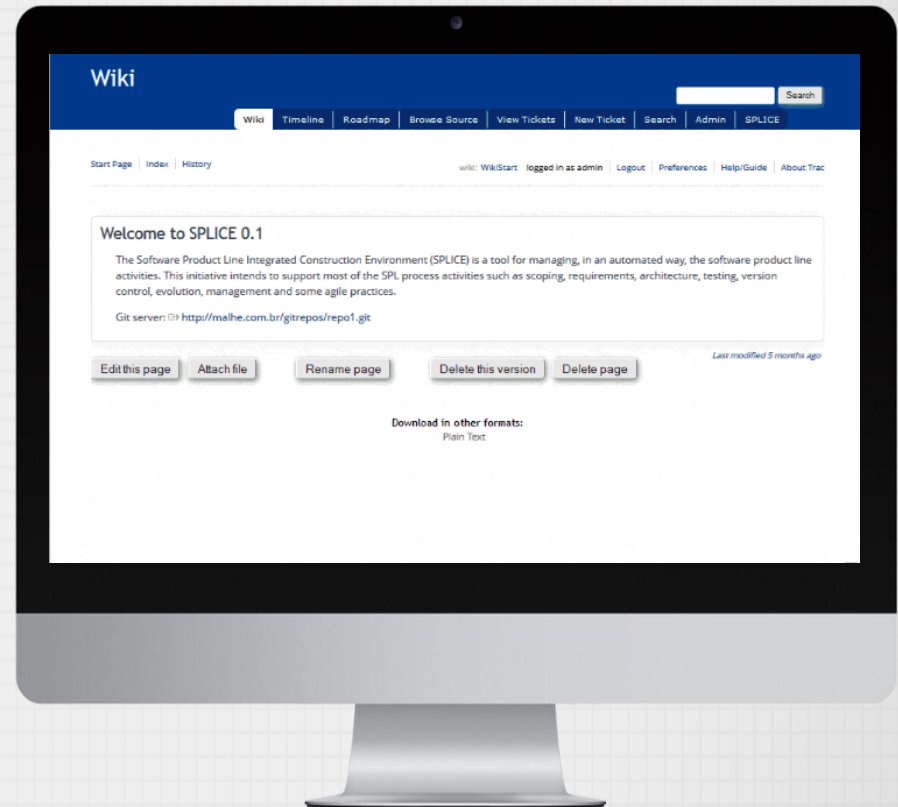


- **SEGURANÇA**
- Por ser um sistema publicamente acessível, deve possuir um controle de acesso rigoroso e adotar medidas para preservar a confidencialidade.

REQUISITO: DOCUMENTAÇÃO COLABORATIVA

DESCRIÇÃO

Foi integrado um sistema de documentação colaborativa “Wiki”. Wiki possui uma linguagem simples, e recentemente tem ganhado popularidade para publicação de conteúdo. Bom para gestão de conhecimento.



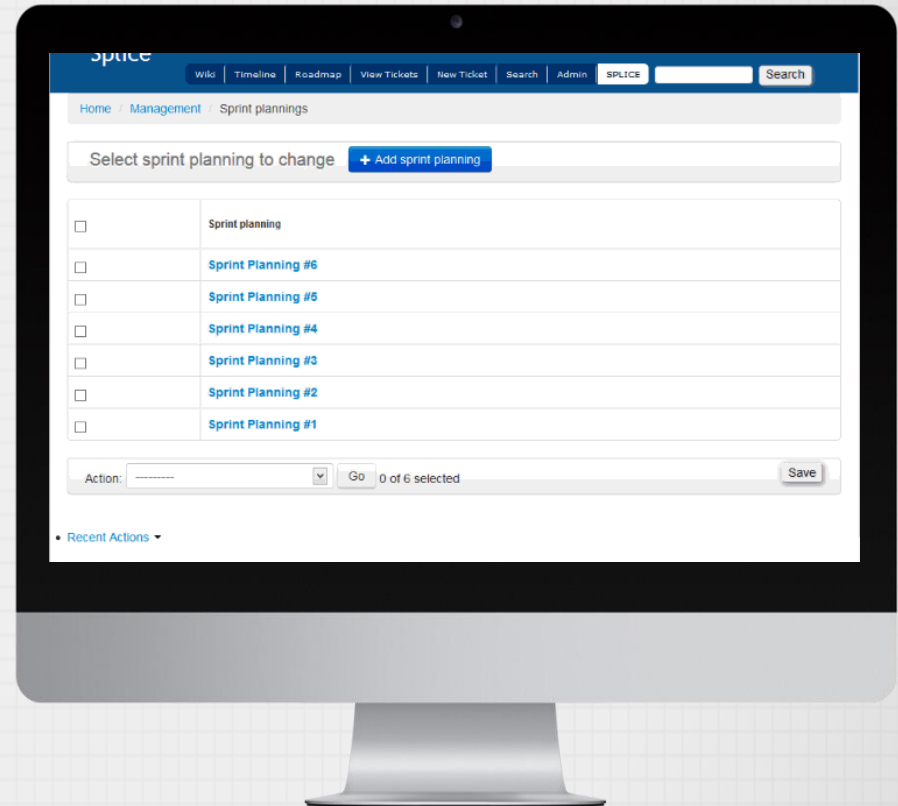
REQUISITO: PLANEJAMENTO ÁGIL

DESCRIÇÃO

Existe um “trend” na indústria para adoção de praticas ágeis.

Alguma das características inclui:

- Participação do consumidor
- Equipe auto organizáveis
- Ênfase na codificação
- Burocracia mínima
- Entrega incremental
- Desenvolvimento baseado em testes

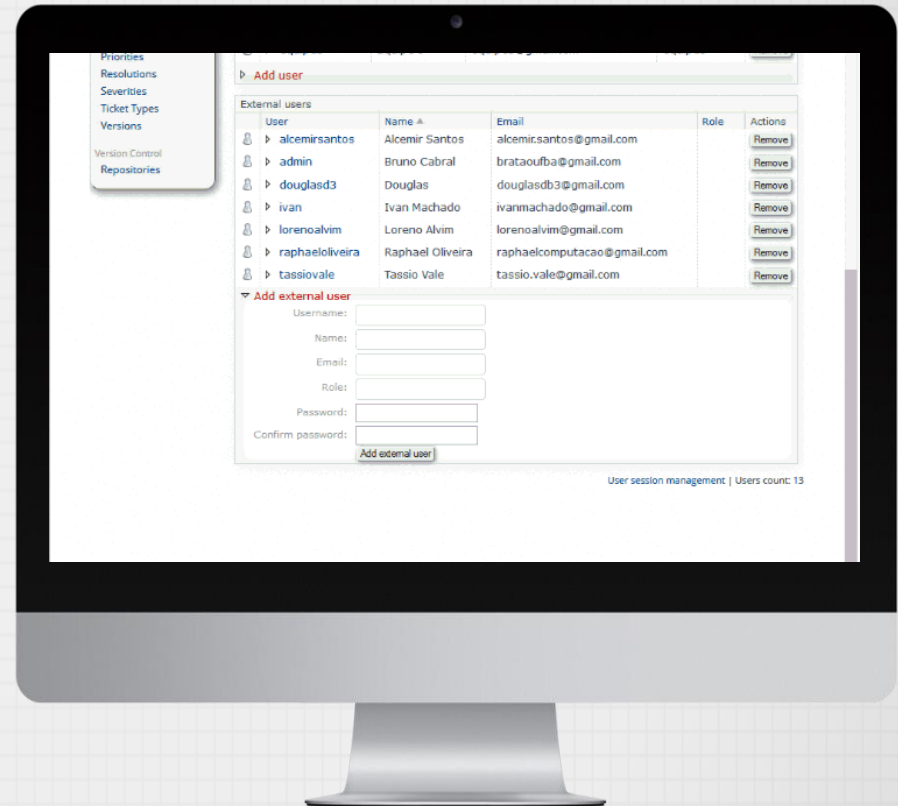


REQUISITO: GERENCIAMENTO UNIFICADO DE USUÁRIOS

DESCRIÇÃO

O SPLICE é um ambiente integrado que inclui uma série de ferramentas externas.

Por conveniência, o usuário deve controlar as contas de usuários e permissões por uma interface única.

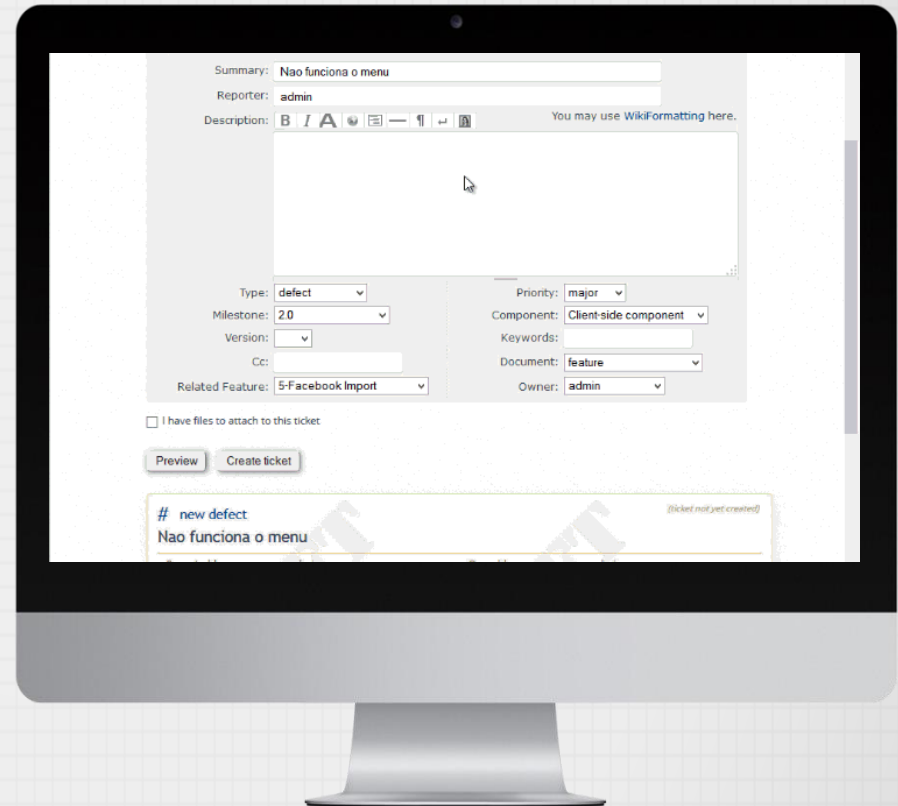


REQUISITO: SISTEMA DE TICKETS

DESCRIÇÃO

Gerenciamento de Tickets em muitos projetos é uma atividade essencial para correção e acompanhamento de bugs, e implementação de novas atividades.

Temos um sistema completo de tickets.



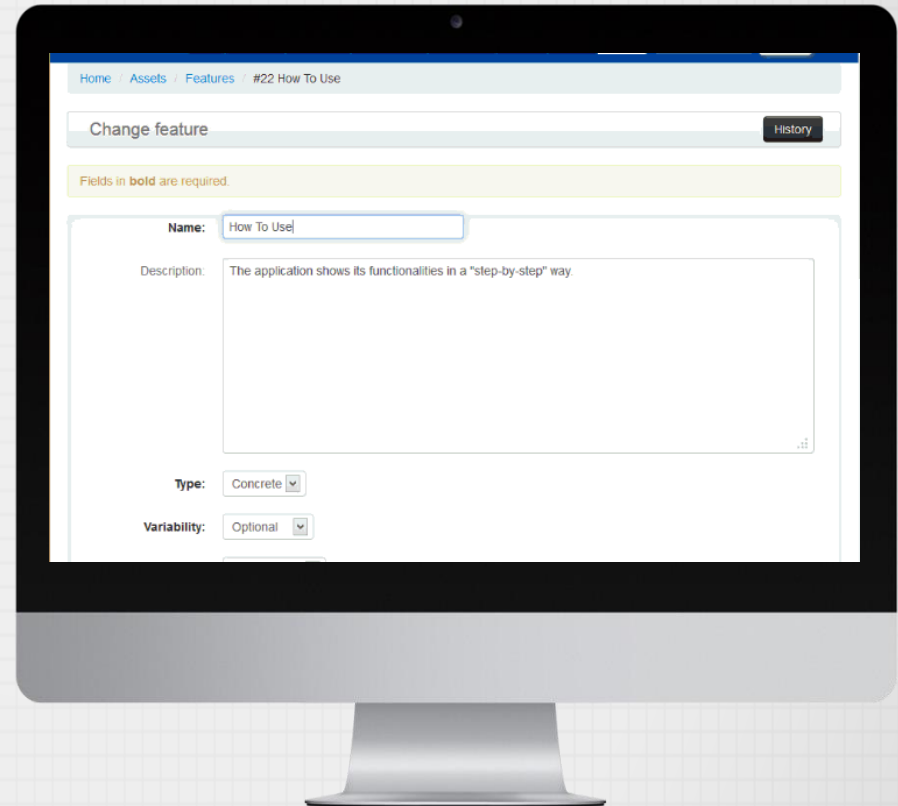
REQUISITO: GERÊNCIA DE CONFIGURAÇÃO

23

DESCRIÇÃO

Para gerenciar a evolução, todos os artefatos precisam ter um mecanismo de controle de mudança.

Precisa também prover suporte e controlar os principais sistemas de controle de versão do mercado (SVN; GIT)

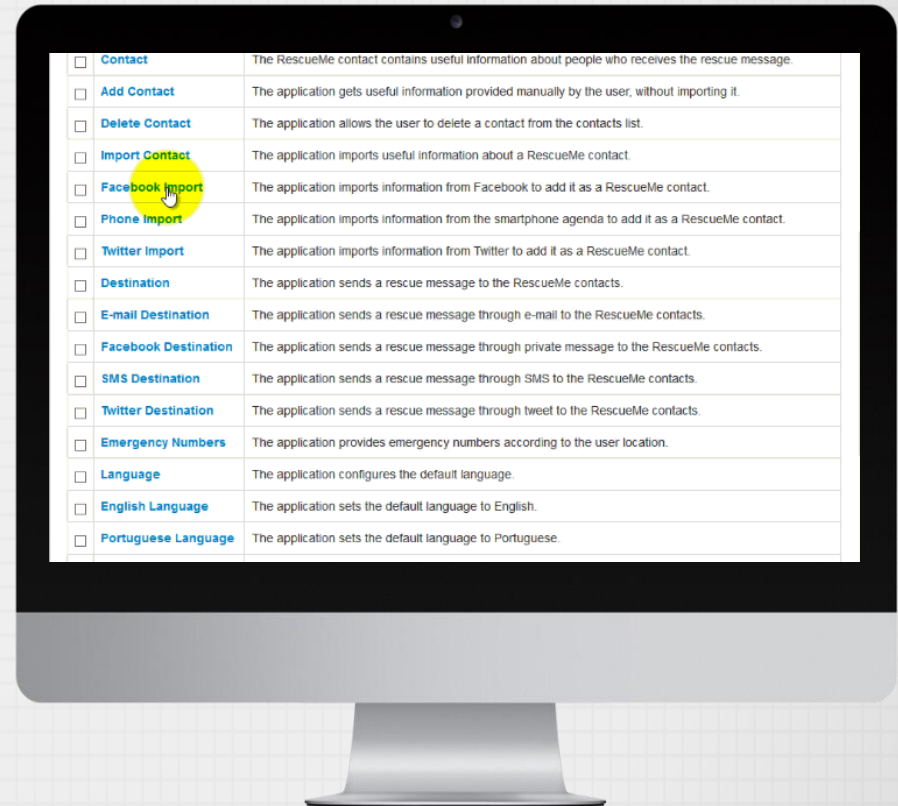


REQUISITO: RASTREABILIDADE

DESCRIÇÃO

A ferramenta deve identificar e manter todas as relações entre os artefatos gerenciados.

A rastreabilidade é importante não apenas para a geração de relatórios, mas também para análise do impacto de mudanças, e visualização durante o desenvolvimento.

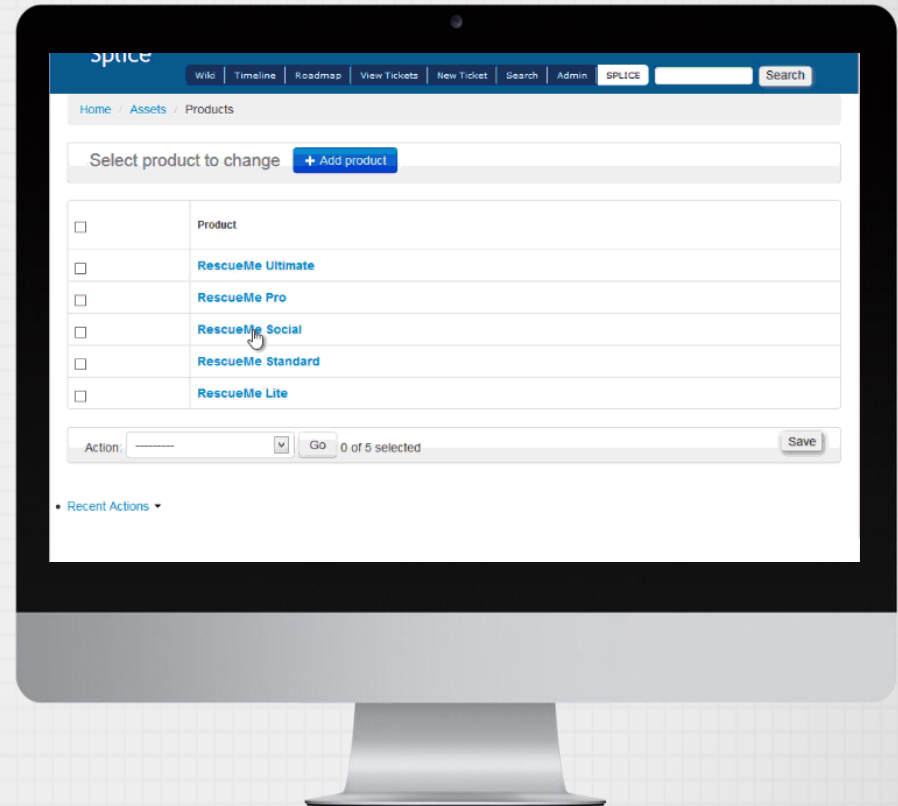


REQUISITO: GERAÇÃO DE RELATÓRIOS

DESCRIÇÃO

Deve utilizar os artefatos e as informações de rastreabilidade para geração de relatórios.

Os relatórios gerados podem ser diretamente consumidos pelos Stakeholders.



ESTUDO DE CASO : RESCUEME

Migração de um processo manual para o SPLICE.



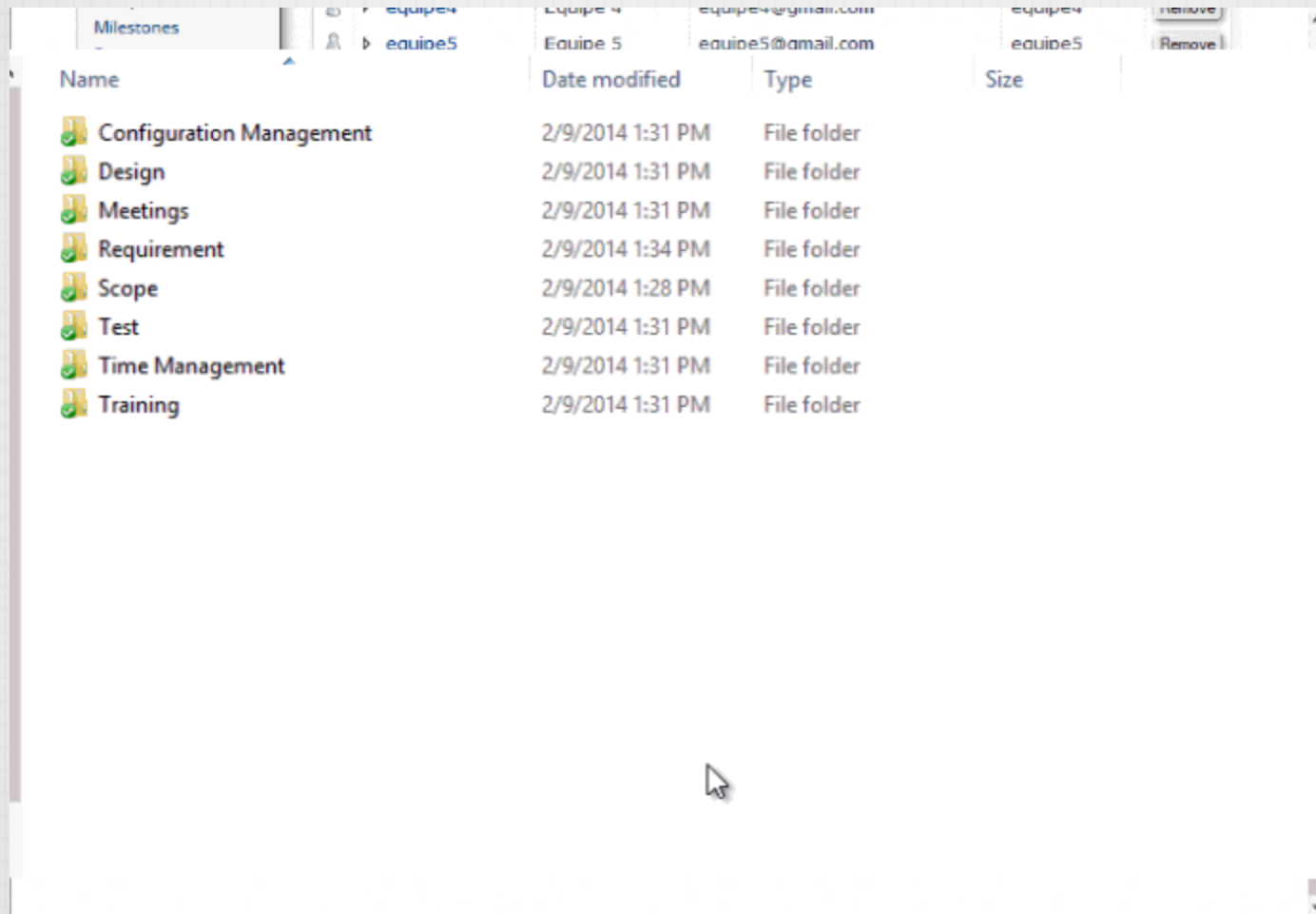
CONTEXTO

Durante os meses de junho e novembro de 2013, foi realizado um estudo de caso no "Instituto Nacional de Engenharia de Software (INES)," um laboratório de pesquisa de Engenharia de Software, composto por 11 candidatos a Ph.D

PRODUTO

O laboratório desenvolveu um LPS chamado RescueMe, que foi construído seguindo um processo LPS ágil. O RescueMe é uma linha de produtos desenvolvidos em Objective-C para dispositivos iOS. RescueMe é projetado para ajudar os seus usuários em situações perigosas.

PROCESSO ANTES DA SPLICE



Name	Date modified	Type	Size	
Configuration Management	2/9/2014 1:31 PM	File folder		
Design	2/9/2014 1:31 PM	File folder		
Meetings	2/9/2014 1:31 PM	File folder		
Requirement	2/9/2014 1:34 PM	File folder		
Scope	2/9/2014 1:28 PM	File folder		
Test	2/9/2014 1:31 PM	File folder		
Time Management	2/9/2014 1:31 PM	File folder		
Training	2/9/2014 1:31 PM	File folder		

COLETA DE DADOS

COMO FERRAMENTA DE COLETA DE DADOS, UTILIZAMOS QUESTIONÁRIOS QUE ADMINISTRAMOS PARA EXPERTS.

NOME	OCUPAÇÃO	EXPERIÊNCIA COM E.S	EXPERIÊNCIA COM L.P.S
Raphael Oliveira	Estudante de Doutorado	10 anos	6 anos
Tássio Vale	Estudante de Doutorado	6 anos	4 anos com experiência na indústria

O questionário foi composto de 3 questões pessoais, oito questões fechadas com campo para justificativa, e três questões abertas

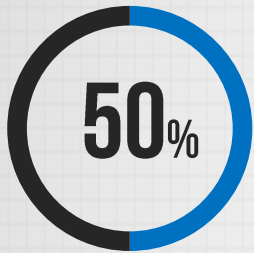
PERGUNTAS

Da perspectiva do stakeholder, como a rastreabilidade é resolvida pela SPLICE ?

Quão positivamente a SPLICE impactou o ciclo de vida ?

Quão negativamente a SPLICE impactou o ciclo de vida ?

RESULTADOS



**DIFICULDADES USO
DA FERRAMENTA**



**DIFICULDADES GERAÇÃO
DE ARTEFATOS**



**PROBLEMAS
USABILIDADE**



**AJUDOU NA
RASTREABILIDADE**

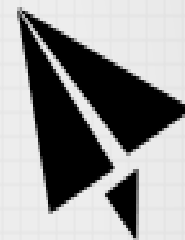


**USARIAM A
FERRAMENTA**

SUGESTÕES

- Dificuldade de modificar o metamodelo
- Melhor análise de impacto
- Integração com o código fonte para realizar a derivação da linha de produto

MUITO OBRIGADO



REFERÊNCIAS

Software Architecture: Foundations, Theory, and Practice; Richard N. Taylor, Nenad Medvidovic, and Eric M. Dashofy; (C) 2008 John Wiley & Sons.

Gears, Biglevel Software, Inc

The noun Project. <http://thenounproject.com/>

Software Product Lines Essentials – Linda

Pohl, K., Bckle, G., van der Linden, F.J.: *Software Product Line Engineering: Foundations, Principles*

Ingrid Oliveira de Nunes, PUC-Rio, Departamento de Informática, LES

Clements, P., Northrop, L.: *Software Product Lines: Practices and Patterns*. Addison-Wesley, Boston, MA, USA (2002).

Software Product Lines. <http://www.sei.cmu.edu/productlines/>