

ARTIGO TÉCNICO

Arquitetura de Negócio Estratégica: Planejamento & Roteirização no Horizzon

Marc Lankhorst
Sven van Dijk



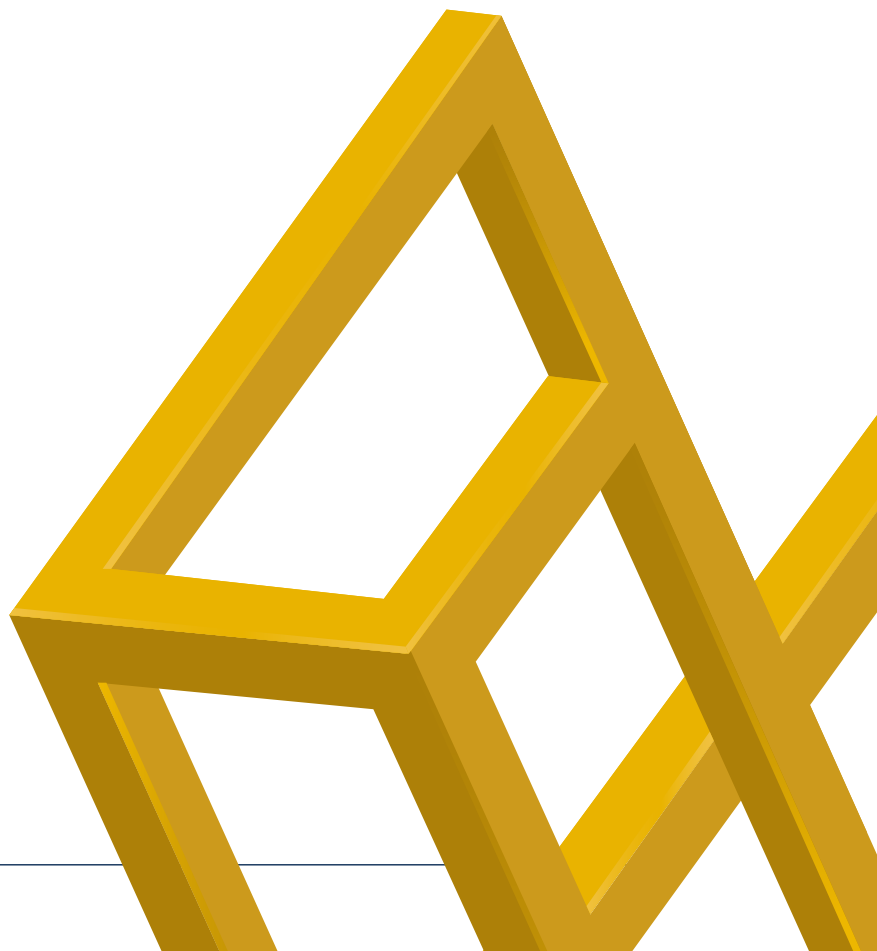
Bizzdesign



Introdução

Um dos principais motivadores para a gestão em geral e para a arquitetura corporativa, em particular, é obter um melhor controle sobre o futuro, sobre a evolução da sua empresa. Uma técnica comum usada para suportar este objetivo é a roteirização. Um roteiro é um plano estratégico que mostra os principais passos ou marcos necessários para alcançar um resultado desejado. Ele articula a direção estratégica da sua empresa e mostra o caminho adiante. Ele ajuda você a identificar o que é necessário e quais são as principais dependências e prioridades, e serve como um instrumento de comunicação para alinhar a organização em torno de um curso de ação comum.

É importante perceber que quanto mais no futuro nós planejamos, menos concretos os nossos planos podem, e devem, necessariamente ser. Por um lado, nós só podemos explorar e desenvolver os conhecimentos necessários no decorrer do processo, porque, por exemplo, nossos usuários só podem nos dizer as suas necessidades de uma maneira 'eu sei isso quando eu vejo isso'. Por outro lado, o mundo continua a mudar em torno de nós, portanto temos que ajustar nossos planos de qualquer maneira, tornando quase impossível (e, portanto, um exercício inútil) definir planos detalhados distantes no futuro. Um roteiro não é, certamente, um plano rígido para os próximos cinco anos. Trata-se de um instrumento flexível para dar direção para a sua empresa e identificar as principais dependências. A grande figura e as mais partes distantes de seu roteiro serão, portanto, pintados usando um pincel mais amplo.



Planejamento Baseado em Capacidades & Roteirização

Um importante caso de uso para os roteiros é o planejamento baseado em capacidades. A evolução das capacidades em termos dos assim chamados "incrementos" é algo que você pode perfeitamente expressar através de um roteiro. A Figura 1 mostra um roteiro simples em que a capacidade de Gerenciamento de Clientes é gradualmente melhorada através de vários projetos para homogeneizar a informação e os dados sobre os clientes através das unidades de negócio. Isso melhora o acesso global àqueles dados e harmoniza os processos de faturamento através das várias unidades de negócio.

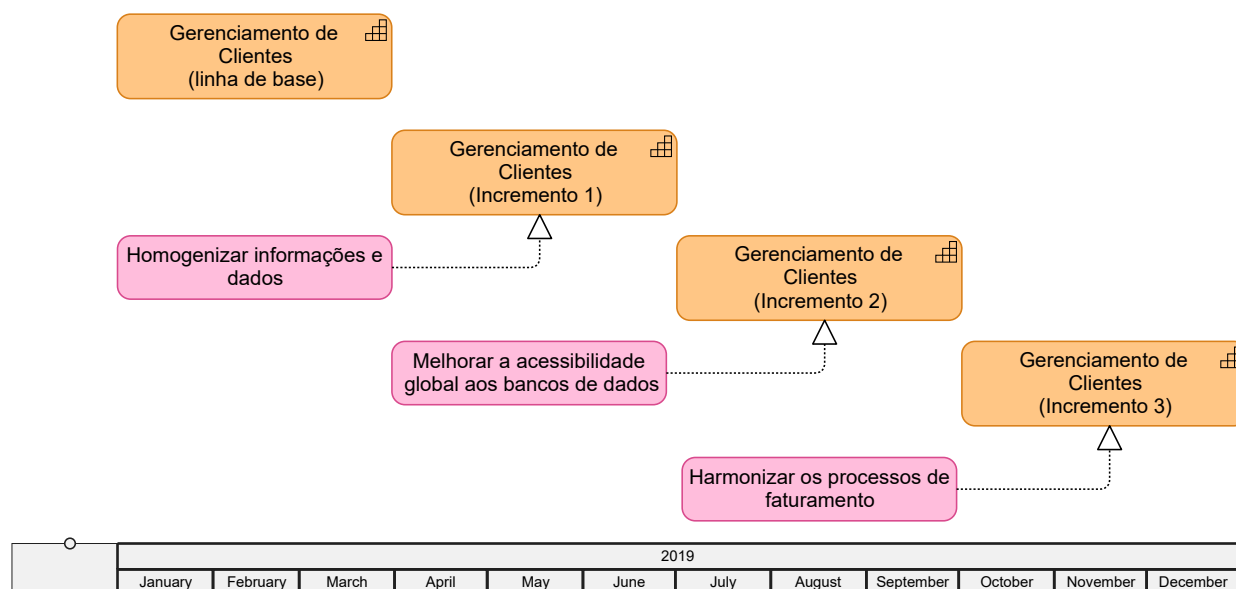


Figura 1: Planejamento Baseado em Capacidades

Os projetos serão detalhados em unidades menores de trabalho, e os incrementos de capacidade em si exigirão melhorias específicas em vários aspectos da arquitetura. Estas melhorias podem ser desenvolvidas através de pacotes de trabalho específicos no âmbito destes projetos e expressos pelos seus resultados apropriados.

Roteiros Multinível

Os roteiros podem ser definidos de várias maneiras e em vários níveis. No contexto do desenvolvimento ágil de software, por exemplo, no **Scaled Agile Framework**, muitas vezes você verá vários níveis e escalas de tempo, tais como:

- Roteiro: iterações de vários meses ou mesmo anos.
- Release: iterações de poucos meses.
- Sprint: iterações de algumas semanas.

Nós consideramos tudo isso partes do 'roteamento' como uma técnica geral.

A evolução das suas capacidades pode ser expressa em termos de funcionalidades e histórias, e colocadas no backlog de algum sprint, a ser desenvolvido pelas várias equipes ágeis disponíveis. Vários sprints de várias equipes ágeis em conjunto resultam em um release que fornece um incremento de capacidade, e uma série de releases em conjunto, por sua vez, pode resultar em um resultado específico de negócio.

Ao criar um roteiro multinível, você definirá o conteúdo pormenorizado dos sprints apenas no curto prazo, com histórias de usuário específicas para serem implementadas e mudanças concretas nos processos e sistemas. Releases serão planejados talvez alguns poucos meses no futuro, mas as funcionalidades para cada release serão especificadas de uma maneira de mais alto nível e com maior granularidade, e serão detalhadas somente para o release mais próximo. Fases do roteiro em um nível mais elevado e escala de tempo mais longa serão ainda mais abstratas, por exemplo, identificando as melhorias de capacidades (como no exemplo acima), ou mesmo apenas dando uma direção na forma de metas de longo prazo e resultados desejados.

Modelando Roteiros no ArchiMate

A característica mais marcante da linguagem ArchiMate é o conceito de Platô. No padrão, este é definido como: “um estado relativamente estável da arquitetura que existe durante um período limitado de tempo”. A intenção deste conceito é modelar alterações razoavelmente grandes em sua arquitetura, por isso o “relativamente estável”; mudanças de pequena escala são demasiado frequentes para serem facilmente reunidas em platôs.

Por exemplo, você não gostaria de modelar cada correção individual de software desta forma. Uma vez que, de qualquer maneira, estas raramente têm significado arquitetural (mesmo que sejam importantes por outras razões, é claro), um modelo de arquitetura não é o melhor lugar para captar tais mudanças. Por isso, essa forma de modelagem é particularmente útil nos níveis estratégicos e táticos em que a arquitetura corporativa normalmente está preocupada.

Um platô é válido por algum período de tempo, de modo que ele normalmente tem uma data de início e fim. Um platô pode entrar em vigor depois que algum projeto ou trem de entrega ágil implemente uma mudança, e permanecer em vigor até que a próxima mudança aconteça. Em termos de relacionamentos ArchiMate, platôs podem agregar qualquer elemento ou relacionamento na arquitetura que é atual no seu período de tempo designado.

Seu escopo arquitetural pode ser limitado a uma parte específica da arquitetura, digamos um domínio de negócio ou um sistema específico. Vários sub-platôs podem ser agregados em um platô maior, digamos para modelar o roteiro global que consiste na evolução dos vários domínios de negócio. Naturalmente, você deveria, então, manter os períodos de tempo dos sub-platôs coerentes com seu platô-pai, caso contrário o modelo não fará muito sentido.

Modelando Cenários

Você também pode usar platôs para expressar caminhos alternativos em direção ao futuro. A Figura 2 dá um exemplo disso, em um cenário após a fusão de duas organizações: em Transição A, um sistema de CRM comum é implementado em primeiro lugar, enquanto em Transição B os sistemas de retaguarda são substituídos primeiro. Transição C é, então, o estado em que ambos, o CRM e os sistemas de retaguarda, foram consolidados.

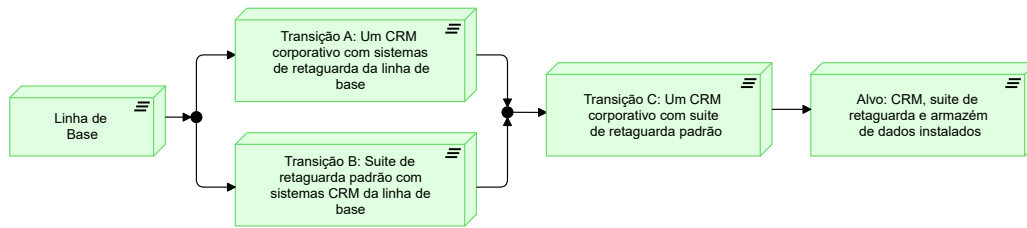
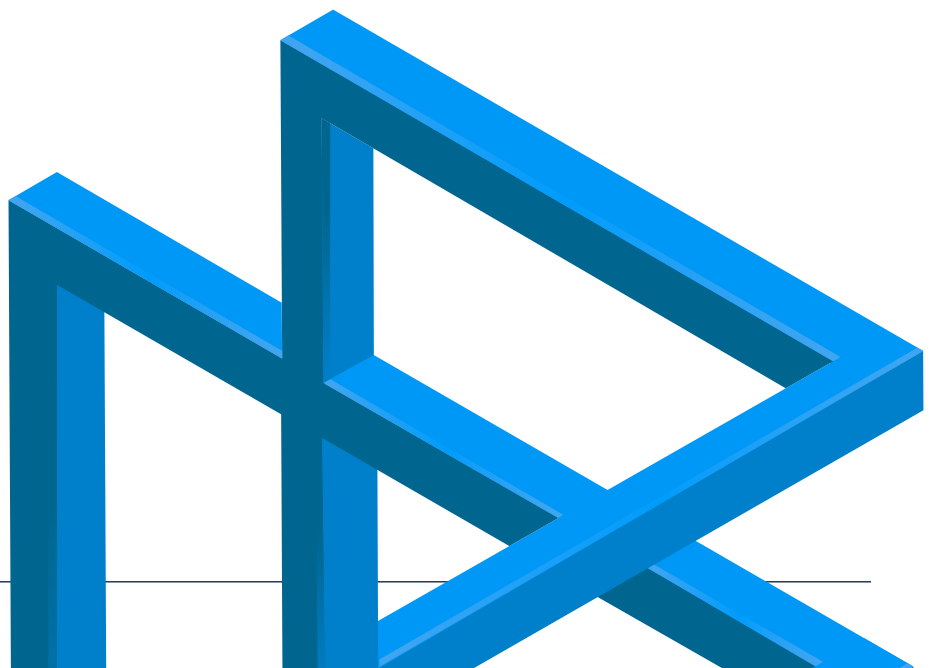


Figura 2: Roteiro com platôs

Adicionando Pacotes de Trabalho à mistura

Ao lado dos Platôs, Pacotes de Trabalho são usados para modelar o trabalho a ser feito nos diferentes níveis envolvidos, desde os sprints individuais até os roteiros completos. A evolução resultante da arquitetura é capturada em Platôs, que agregam os elementos da arquitetura válidos durante um determinado período de tempo. A Figura 3 mostra um pouco disso. À esquerda, vemos o desenvolvimento dos processos, sistemas e dados concretos, organizados em sprints modelados através dos conceitos de Platô e Pacote de Trabalho do ArchiMate. Várias equipes podem estar trabalhando durante o mesmo período de sprint, por isso a replicação dos Pacotes de Trabalho.

À direita, vemos incrementos de capacidade mais abstratos organizados em releases que são, eles mesmos, o resultado de uma série de sprints. No modelo subjacente (mas não mostrado na figura), todos os elementos criados ou alterados nos dois sprints contribuem para o primeiro incremento de capacidade. Note que, ao contrário da Figura 1, esta não retrata uma linha do tempo da esquerda para a direita.



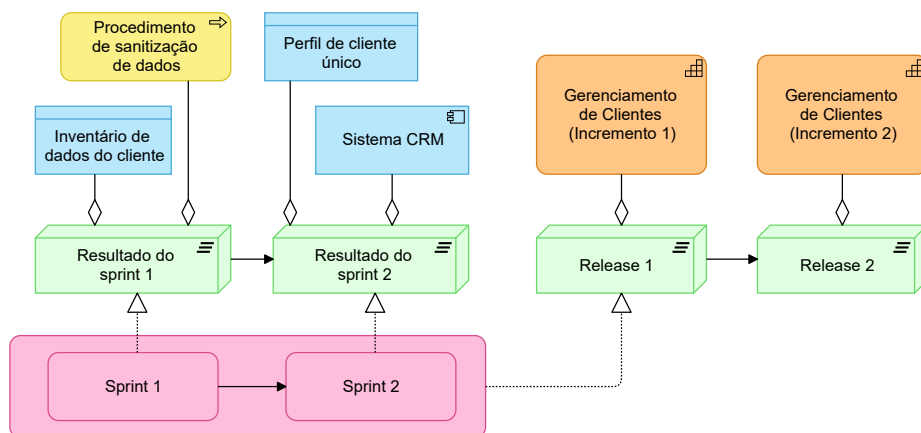


Figura 3: Estágios do roteiro de alto nível e detalhados

Como mencionado acima, quanto mais você olhar para a frente, menos concreto o seu roteiro será. Assim, os conceitos utilizados para o futuro próximo são, normalmente, os elementos concretos e detalhados das camadas de negócio, aplicativo e tecnologia. Para o futuro de médio prazo, você poderia utilizar os elementos mais abstratos de capacidade e recurso e, a longo prazo, talvez você expressasse isso utilizando apenas os conceitos de meta e resultado.

A notação gráfica para platôs é útil principalmente para mapear uma evolução como essa, em vez de organizar a arquitetura em si, uma vez que muitos elementos em sua arquitetura farão parte de vários platôs. Por exemplo, qualquer coisa na linha de base que sobreviva através das transições acima pode ser agregada em todos os platôs. Por essa razão, nós frequentemente usamos platôs em um modelo de arquitetura, mas não necessariamente em uma visão como a mostrada na Figura 3.

Em vez disso, nós entramos os relacionamentos de agregação entre platôs e seu conteúdo através de uma tabela de referência cruzada, e usamos isso nas visões e análises. A Figura 4 mostra um exemplo disso, destacando os elementos de acordo com os platôs em que eles se encontram, com base no mesmo cenário mostrado acima.

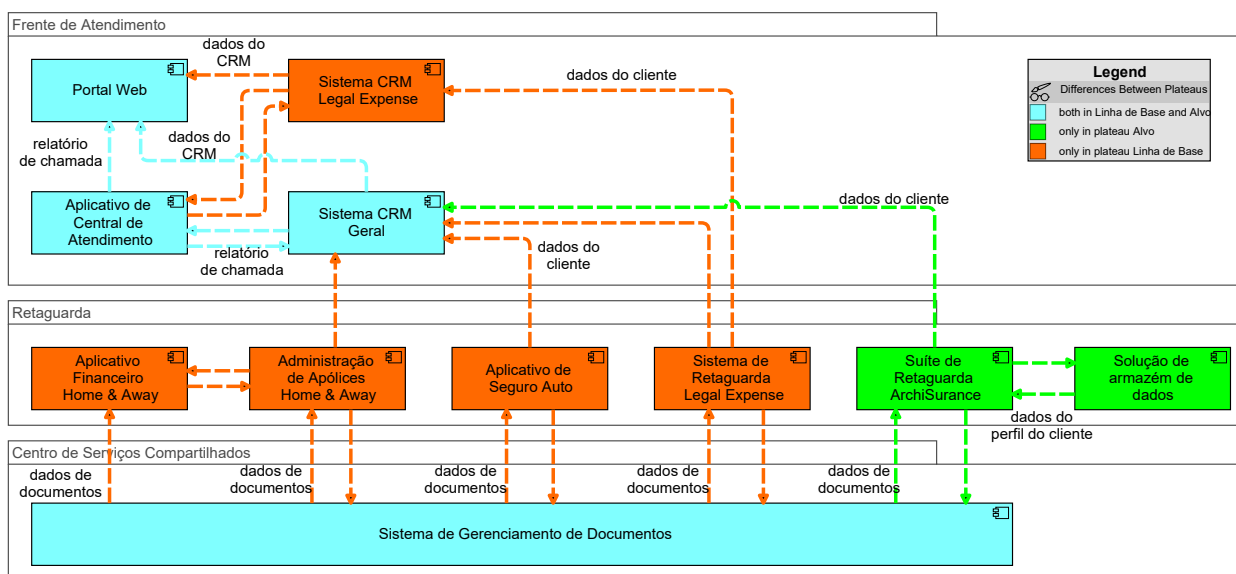


Figura 4: Destacando platôs na arquitetura

Até agora, discutimos a ideia geral de planejamento e roteirização no contexto da arquitetura corporativa e do planejamento baseado em capacidades. Nós abordamos diferentes níveis de roteiros, que variam de sprints de curto prazo de algumas semanas, a roteiros de longo prazo e cobrindo vários anos. Nós também fornecemos algumas percepções iniciais para o uso de conceitos ArchiMate para a modelagem de roteiros.

Nas próximas três seções, vamos explorar algumas maneiras diferentes de modelar a evolução de sua arquitetura com um pouco mais de profundidade através do uso de conceitos ArchiMate para diferentes prazos e níveis de detalhe.

Modelando Incrementos & Releases como Especializações

Além da evolução de alta granularidade dos roteiros descrita acima, você pode também querer modelar que algum elemento da arquitetura é melhorado. Um exemplo comum disso é usar "incrementos de capacidade" para expressar que a maturidade de alguma capacidade cresce ao longo do tempo.

Uma boa maneira de fazer isso é modelar cada incremento como uma especialização de um elemento genérico de nível superior, neste caso aquela capacidade específica, e agregar essas especializações nos platôs apropriados. Desta forma, você pode ver tanto a estrutura geral da arquitetura, como, por exemplo, quais elementos são necessários para realizar essa capacidade, e mostrar também as melhorias consecutivas. O mesmo padrão funciona para a modelagem de releases de software.

Por exemplo, conforme mostrado na Figura 5, você pode modelar o elemento genérico "PaymentApp" com especializações como 'PaymentApp 3.3', 'PaymentApp 3.4' e 'PaymentApp 4.0'. Cada uma destas especializações, por sua vez, pode estar relacionada a um platô que começa na data de liberação ou implantação dessa versão e termina quando ela é substituída pela próxima.

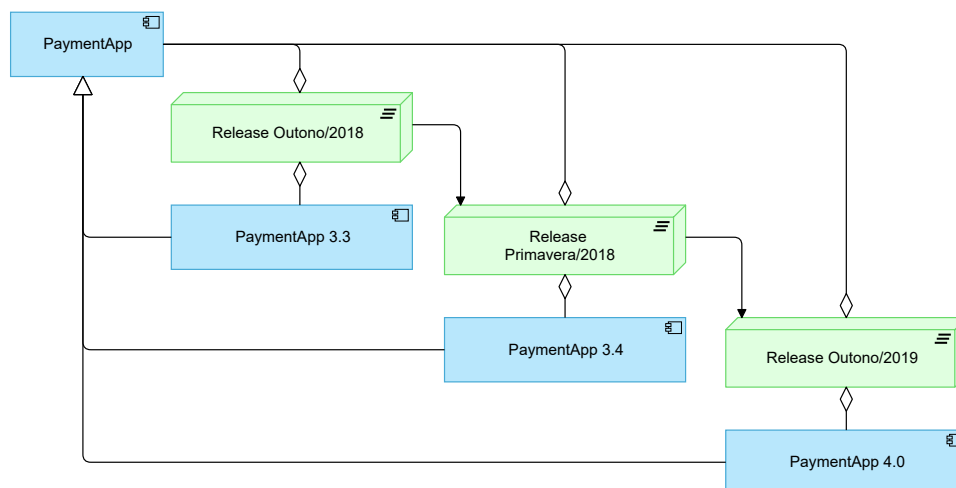


Figura 5: Roteiro com releases e versões

Modelando Ciclos de Vida

Agora, vamos dirigir nossa atenção em direção a modelar o tempo de forma mais detalhada e em nível fino no Enterprise Studio. Especificamente, vamos ver como você pode modelar ciclos de vida de elementos individuais e usar as dependências entre os estágios do ciclo de vida de diferentes elementos em sua arquitetura para identificar, por exemplo, conflitos em seus planos de transição.

No Enterprise Studio, você pode adicionar propriedades temporizadas aos elementos do seu modelo. Como o nome diz, o valor de tais propriedades pode mudar ao longo do tempo. Basta pensar, por exemplo, em propriedades que representam a versão de um componente de aplicativo ou o nível de maturidade de uma capacidade. Propriedades temporizadas podem ser usadas para todos os tipos de propósito em nossa plataforma, mas aqui vamos usá-las para a modelagem de ciclo de vida.

O Enterprise Studio vem com vários ciclos de vida pré-configurados para diferentes tipos de elementos, tais como aplicativos e projetos, para citar alguns. Ao lado desses, você também pode definir os seus próprios ciclos de vida. Na Figura 6 abaixo, você pode ver uma visão de ciclo de vida de um conjunto de aplicativos, com todos os ciclos de vida individuais plotados em uma linha do tempo.

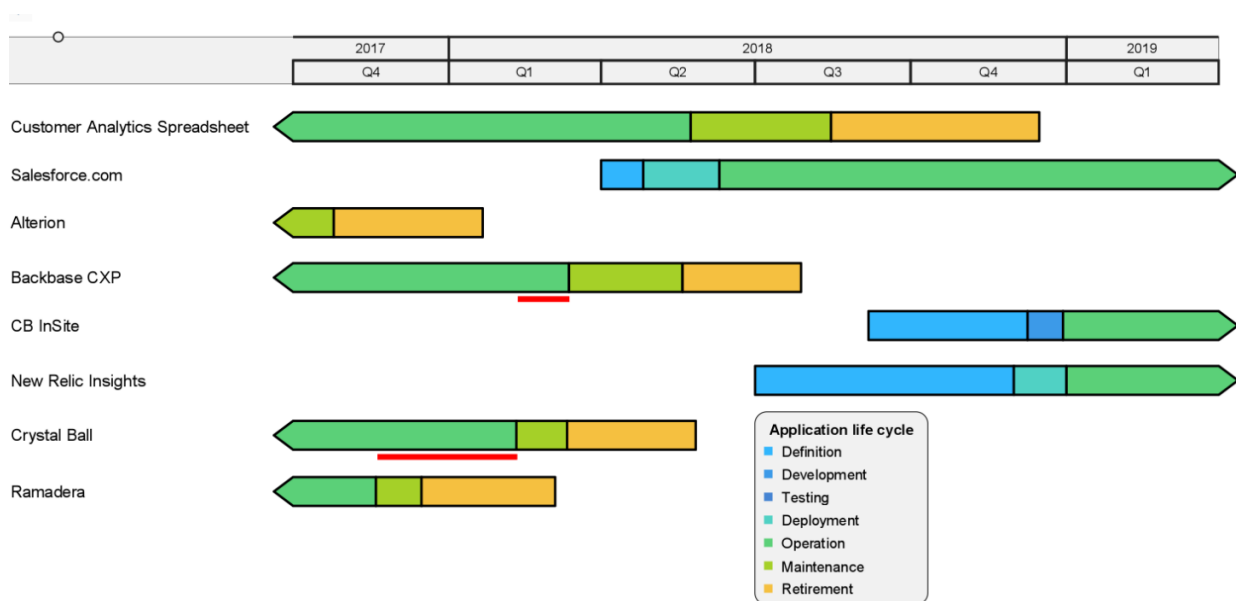


Figura 6: Visão de ciclo de vida

Nesta visão, talvez você também perceba as barras vermelhas abaixo de dois aplicativos. Elas denotam conflitos de ciclo de vida, como, por exemplo, eles podem estar trocando dados ou usando funcionalidades um do outro. Clicar na barra vermelha irá destacar o aplicativo "ofensivo".

Na Figura 7, a visão ArchiMate mostra exatamente como os aplicativos no exemplo anterior estão relacionados uns aos outros na arquitetura. Destacamos os relacionamentos que causam os conflitos de ciclo de vida mostrados acima: Ramadera serve Crystal Ball, que, por sua vez, serve Backbase CXP. Assim, se Ramadera fica fora do ar porque está em manutenção, Crystal Ball não pode fazer o seu trabalho.

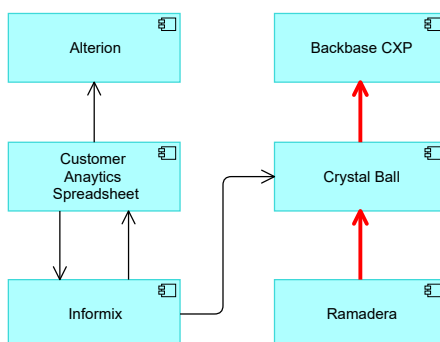


Figura 7: Panorama parcial de aplicativos, destacando dependências causando conflitos de ciclo de vida

Como os diferentes relacionamentos em sua arquitetura afetam as dependências de estágios do ciclo de vida é totalmente configurável. Isto pode incluir também dependências "através das camadas" como, por exemplo:

- processos de negócio que utilizam serviços e componentes de aplicativo
- aplicações que utilizam determinada tecnologia, por exemplo, sistemas operacionais, interfaces, ou plataformas de banco de dados
- projetos que realizam várias partes da arquitetura

Análises de Ciclo de Vida Mais Avançadas

Você também pode fazer estas análises para casos mais complexos, tais como relacionando os estágios de um projeto com o ciclo de vida dos aplicativos que eles constroem, ou determinando o que acontece se uma equipe ágil decide adiar uma funcionalidade para o sprint seguinte. Isso pode ajudar você para avaliar quais funcionalidades relevantes para o negócio terão que ser adiadas e quais clientes poderiam ser afetados. Com base nesta informação, você pode, então, decidir sobre as prioridades do desenvolvimento.

A Figura 8 mostra um exemplo mais elaborado. No topo, vemos como um conjunto de pacotes de trabalho depende das saídas de outros pacotes de trabalho. Neste cenário, estes pacotes de trabalho refletem vários projetos, mas eles poderiam também ser usados para modelar, por exemplo, sprints em um contexto ágil. O sinal de aviso na figura e a tabela à esquerda mostram que se o projeto P428 terminar muito tarde um conflito ocorre, uma vez que o Projeto P473 precisa da saída deste projeto anterior.

Mais ainda, existem problemas potenciais com uma sobreposição de escopo, como mostrado na tabela à direita. Os projetos P431 e P473 trabalham ambos no aplicativo New Relic Insights. Se eles também se sobrepõem no tempo, um pode acabar pisando no pé do outro.

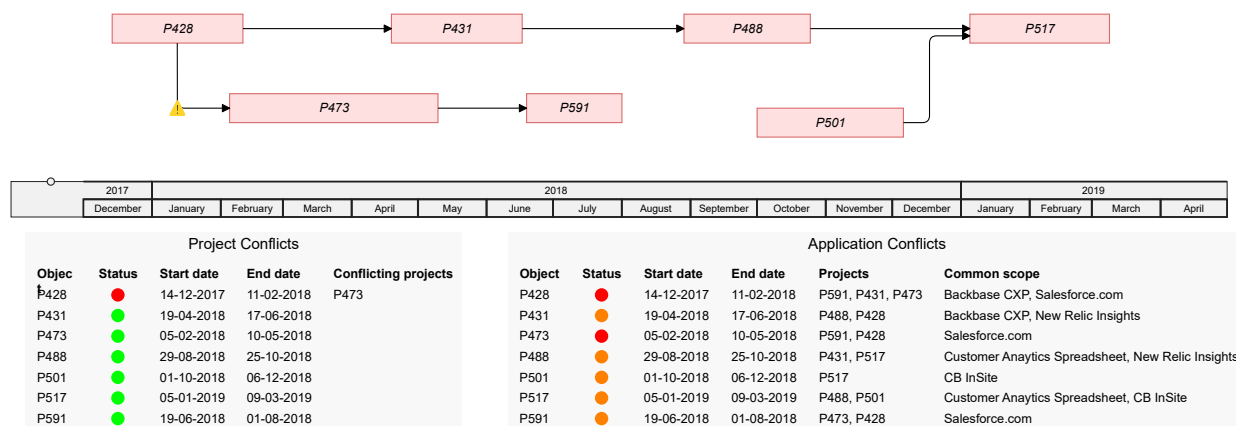


Figura 8: Planejamento e dependências de projetos

A informação sobre o escopo dos projetos é, por sua vez, obtida em outra parte do modelo. A Figura 9, abaixo, demonstra o roteiro de alto nível para este programa de racionalização de aplicativos. As caixas rosas representam os mesmos projetos como acima, movendo dados do aplicativo antigo para o novo. Se um projeto toca um aplicativo, aquele aplicativo está no escopo daquele projeto.

Este roteiro usa o conceito de Platô do ArchiMate para modelar os estágios principais na transição do panorama de aplicativos, de uma forma similar à que nós ilustramos anteriormente. Assim, você pode conectar seu planejamento de alto nível, grosseiro, com as informações detalhadas sobre os ciclos de vida dos aplicativos, projetos e outros elementos individuais.

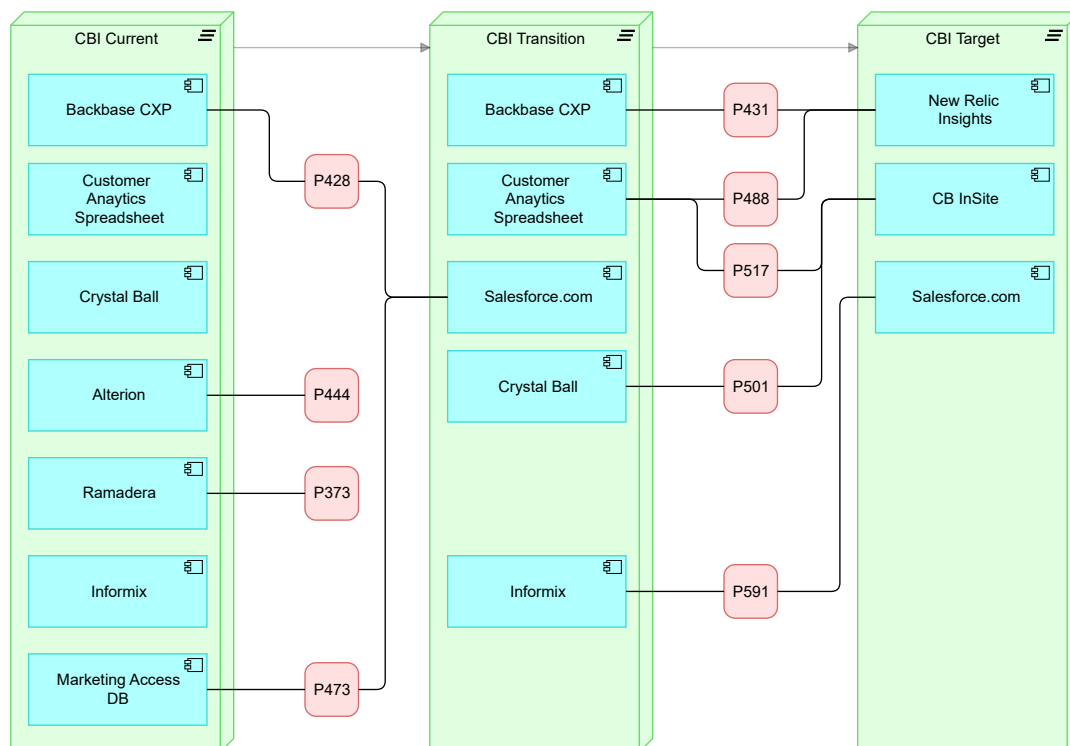


Figura 9: Roteiro de alto nível

Se você quer saber mais sobre estas funcionalidades avançadas para o planejamento e roteirização no Enterprise Studio, fique à vontade para **solicitar uma demonstração!**

Sobre a Bizzdesign

Fundada em 2000, a Bizzdesign é uma plataforma SaaS global e confiável de Arquitetura Corporativa reconhecida como um líder pelas principais empresas de análise de mercado. Nós ajudamos organizações públicas e privadas líderes mundiais para garantir o sucesso da priorização de investimentos, iniciativas de transformação e gerenciamento de riscos. A Bizzdesign ajuda os arquitetos e executivos a ver uma figura completa multidimensional, encontrar e desenhar o caminho certo, e executar com confiança para o seu futuro almejado. O sucesso não deveria ser uma questão de esperança. Ele deveria ser por desenho. Para mais informações, acesse bizzdesign.centus.com.br ou www.bizzdesign.com.

Parceira no Brasil



info@centus.com.br
fone: +55 31 99279-0290
www.centus.com.br