

DAS
CTC
UFSC

Departamento de Automação e Sistemas
Centro Tecnológico
Universidade Federal de Santa Catarina

Análise e Proposição de Melhorias no Processo de Gestão de Projetos e de Desenvolvimento de Software – Um Estudo de Caso.

*Relatório submetido à Universidade Federal de Santa Catarina
como requisito para a aprovação na disciplina
DAS 5511: Projeto de Fim de Curso*

Victor Gomes de Oliveira

Florianópolis, Junho de 2014

**Análise e Proposição de Melhorias no Processo de
Gestão de Projetos e de Desenvolvimento de Software –
Um Estudo de Caso.**

Victor Gomes de Oliveira

Esta monografia foi julgada no contexto da disciplina
DAS5511: Projeto de Fim de Curso
e aprovada na sua forma final pelo
Curso de Engenharia de Controle e Automação

Prof. Ricardo José Rabelo

Assinatura do Orientador

Banca Examinadora:

Marcelo Henrique Salloum dos Santos
Orientador da Empresa

Prof. Ricardo José Rabelo
Orientador do Curso

Avaliador

Debatedores

Agradecimentos

Gostaria de agradecer meus amigos por me apoiarem e acreditarem em mim, meus familiares pelo amor incondicional e meus professores por trazerem luz às minhas ideias.

Sumário

Resumo	7
Abstract	8
Capítulo 1: Introdução	9
Capítulo 2: A Empresa e os Projetos	11
2.1: <i>Cheesecake Labs</i>	11
2.2: <i>Empresas Parceiras</i>	11
2.2.1: Primeiro Caso de Estudo - Triple Point, Cogentio	12
2.2.2: Segundo Caso de Estudo – Camiolog	12
Capítulo 3: Normas, Metodologias e Modelos	14
3.1: <i>Gestão de Projetos</i>	14
3.1.1: Project Management Body of Knowledge (PMBOK)	14
3.1.2: ISO 21500	19
3.1.3: Comparação entre PMBOK e ISO 21500	21
3.2: <i>Gestão de Desenvolvimento Software</i>	25
3.2.1: CMMI	25
3.2.2: MPS.BR	30
3.3: <i>Modelagem</i>	32
3.3.1: Business Process Modeling Notation (BPMN)	32
3.3.2: Unified Modeling Language (UML)	32
3.3.3: IDEF-0	33
Capítulo 4: Modelagem e Geração de Melhorias	35
4.1: <i>Gestão de Projetos</i>	35
4.1.1: Escolha da Metodologia de Gestão e Norma de Modelagem	35
4.1.2: Modelagem do Estado Atual	35
4.1.3: Modelagem do Estado Ideal	41
4.1.4: Proposição e Aplicação de Melhorias	47
4.2: <i>Gestão de Desenvolvimento de Software</i>	58
4.2.1: Escolha da Metodologia	58
4.2.2: Modelagem do Estado Atual	58
4.2.3: Modelagem do Estado Ideal	59
4.2.4: Proposição e Aplicação de Melhorias	60

Capítulo 5: Análise Crítica	64
5.1: <i>Gestão de Projetos.</i>	64
5.1.1: Análise das Ferramentas	64
5.1.2: Avaliação dos Resultados das Melhorias	66
5.2: <i>Gestão de Desenvolvimento de Software</i>	68
5.2.1: Análise da Escolha das Ferramentas	68
5.2.2: Outros aspectos	69
5.3: <i>Análise das Conclusões dos Gestores</i>	69
Capítulo 6: Conclusões	71
Bibliografia:	73

Resumo

Em um contexto global no qual a ascensão do mercado de *smartphones* e *tables* faz com que a procura de engenheiros de *software* nos EUA, em especial no Vale do Silício, dispare, torna-se essencial distribuir o desenvolvimento e gerenciamento de projetos para outros países. No Brasil, o “boom” da web ainda é menor e não aumentou tão significativamente a demanda por engenheiros de *software* nem, conseqüentemente, seus salários. Estabelece-se assim um cenário de desequilíbrio entre oferta e procura por desenvolvedores em diferentes países. A principal barreira que contribui para tal desequilíbrio é a incapacidade de conectar, de maneira produtiva, equipes internacionais trabalhando em projetos inovadores.

A Cheesecake Labs nasceu com o objetivo de desenvolver software e soluções no Brasil para empresas nos EUA tirando, assim, proveito desse grande desnivelamento. Para atingir esse objetivo é essencial estudar como os processos de gestão de projetos e desenvolvimento de software têm que se adaptar para lidar com questões operacionais ímpares: fusos horários dividindo equipes, comunicação limitada pela *web*, constantes mudanças nos conceitos dos produtos e complexos problemas de lógica, escalabilidade e melhores práticas.

Este trabalho, compreendendo o cenário técnico e acadêmico, busca:

- Estudar as formas mais consagradas de gestão de projetos e processos de desenvolvimento de *software*.
- Modelar, analisar, propor e implementar melhorias em dois projetos internacionais da Cheesecake Labs.
- Avaliar, através da visão do gestor da empresa, os primeiros resultados obtidos.
- Avaliar preliminarmente as ferramentas disponíveis na web relacionadas aos temas do projeto e verificar suas vantagens e problemas.

Abstract

In a global context in which the rises of the smartphone and tablet markets boost the demand for software engineers in the U.S., especially in the Silicon Valley, it is essential to hire development and project management work from other countries. In Brazil, the web boom is still smaller and has not increased significantly the demand for software engineers nor, consequently, their salaries. Thus in this a scenario of imbalance between supply and demand by developers in different countries, the strongest barrier that maintains such situation is the inability to connect, productively, international teams working on innovative projects.

The Cheesecake Labs was founded with the goal of developing software and solutions in Brazil for companies in the U.S., thus taking advantage of this market opportunity. To achieve this goal it is essential to study how the processes of project management and software development should be adapted to deal with some complex operational matters: time zones dividing teams, limited communication through the Web, frequent changing concepts of products and complex logic problems, scalability and best practice.

This work, comprises technical and academic development aiming at:

- Studying the most suitable forms of project management and software development processes.
- Modeling, analyze, propose and implement improvements in two international projects in development by Cheesecake Labs.
- Evaluate, through the vision of the manager of the company, the first results obtained.
- Preliminary evaluations of the tools available on the web related to the themes of the project and verify their advantages and problems.

Capítulo 1: Introdução

A empresa Cheesecake Labs, situada em Florianópolis - Santa Catarina, fundada em dezembro de 2013, enfrenta problemas de gestão de projeto e desenvolvimento de software, devido ao grande crescimento de seu corpo de colaboradores (+200% nos últimos 3 meses) e barreiras internacionais. Os problemas da Cheesecake tem consequências críticas em seu funcionamento e previsão de expansão, fazendo com que seu estudo (e aprimoramento) se torne um fator essencial para a evolução da empresa.

A problemática em discussão pode ser dividida em aspectos que envolvem duas áreas de conhecimento:

1. Problemas relacionados aos processos de gestão de projetos:

Os problemas encontrados nos processos de gestão de projetos fazem com que a empresa não tenha precisão, nem controle, de importantes tarefas, como por exemplo: definição de prazos, orçamentos e custos de projetos, correta alocação de recursos humanos, comunicação com clientes, previsão de futuros portfólios, entre outros.

2. Problemas relacionados aos processos de gestão de desenvolvimento de *software*:

Os problemas encontrados na gestão do desenvolvimento de *software* fazem com que a equipe de desenvolvedores tenha dificuldade em: criar soluções de alta qualidade, garantir correto funcionamento do produto em situações críticas, controlar o estado de servidores, cumprir prazos, entre outros.

É importante salientar que alguns dos problemas não pertencem apenas a uma categoria, pois sua solução é multidisciplinar. Neste documento iremos simplificar o contexto para que o estudo das normas e metodologias se dê de forma mais objetiva.

Para solucionar os problemas foi feito um estudo das principais ferramentas, normas e metodologias das áreas de interesse. Em seguida, baseando-se nas restrições dos processos da empresa, foram escolhidas quais as referências que melhor se adaptam ao cenário. Em terceira instância, as referências foram aplicadas em dois estudos de caso. Finalmente, foi feita a análise dos resultados dos processos e concluiu-se sobre futuras mudanças.

Os objetivos gerais deste trabalho são: resolver os problemas de gestão de projetos e desenvolvimento de *software* da Cheesecake Labs, melhorar a comunicação da empresa com seus clientes internacionais e definir as "melhores práticas" para guiar o desenvolvimento futuro da organização.

Os objetivos específicos do projeto podem ser divididos em suas duas áreas de atuação:

1. Objetivo no âmbito da gestão de projetos:

Melhorar os processos considerados mais críticos pelo corpo gerencial da Cheesecake Labs: definir a estrutura analítica do projeto (e a sub divisão de tarefas), desenvolver o cronograma e criar o orçamento vinculado ao mesmo.

2. Objetivo no âmbito da gestão de desenvolvimento de *software*:

Foi escolhido, em conjunto com os gerentes da Cheesecake Labs, focar os esforços deste trabalho na criação de testes unitários para auxiliar a gestão de qualidade do produto.

Este trabalho foi organizado de forma a permitir ao leitor conhecer gradualmente a empresa, o contexto do projeto, as tecnologias utilizadas, o embasamento teórico para sua execução, o trabalho desenvolvido e, finalmente, os resultados obtidos na etapa de desenvolvimento.

O presente capítulo é introdutório ao tema e apresenta a estrutura do trabalho e seu contexto. O segundo capítulo apresenta a empresa e suas áreas de desenvolvimento, seguida dos projetos que são utilizados para estudo. O terceiro capítulo ilustra quais as principais normas, metodologias e ferramentas de gestão de projetos e processos desenvolvimento de *software* utilizadas neste trabalho. No quarto capítulo apresenta-se o desenvolvimento efetivo nos dois casos de estudo. O quinto capítulo apresenta a análise dos resultados obtidos e no último capítulo, o documento é concluído através de considerações finais e perspectivas futuras sobre o encaminhamento das mudanças realizadas.

Capítulo 2: A Empresa e os Projetos

2.1: Cheesecake Labs

Os projetos desenvolvidos na Cheesecake Labs tem como clientes e parceiros *startups* norte americanos do vale do silício que buscam terceirizar suas tarefas de desenvolvimento de *software* (ou parte delas). Um fator determinante na relação entre ambas as partes envolvidas é que os contratos possuem cláusulas de transferências de ações, ou seja, os clientes oferecem a opção de compra de suas ações para a Cheesecake labs - após o prazo de carência. O preço das ações no contrato deve estar abaixo da previsão de médio prazo do produto, para que a Cheesecake verifique nas mesmas uma boa oportunidade de negócio.

Os colaboradores da Cheesecake Labs também possuem planos de aquisição de ações da empresa, fazendo com que todos os envolvidos sejam donos de uma fração de todos os produtos. Por serem donos de todos os produtos, os colaboradores sentem-se motivados a realizar suas tarefas com excelência e ajudar os parceiros (de forma a garantir que seus outros projetos, e portanto investimentos, também vão bem).

2.2: Empresas Parceiras

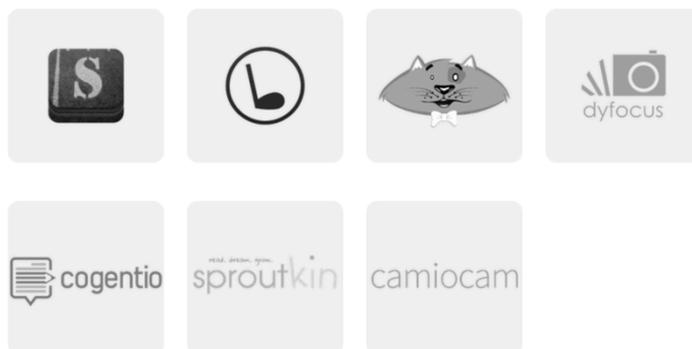


Figura 1 – Logos dos projetos desenvolvidos pela Cheesecake Labs. Fonte:

Autor

Todas as empresas parceiras, para as quais Cheesecake Labs presta serviços, tem um perfil de *startup*, alguns exemplos são:

- Camiocam
Empresa fundada por ex. VP da Google, Carter Maslan.
- Sproutkin

Empresa incubada pelo *500 Startups*.

- Camiolog

Empresa criada em parceria com o fundador e CEO da Triple Point, Richard Kain.

2.2.1: Primeiro Caso de Estudo - Triple Point, Cogentio

A empresa Cogentio foi fundada em 2014 em parceria com a Triple Point (responsável pelas relações públicas de grandes marcas nos EUA, como Warner, Pokémon, BioShock entre outras). O objetivo do produto é realizar o papel de um representante de relações públicas através de um serviço *web*, taxado mensalmente.

A função de um representante de relações públicas é definida pela capacidade do software de conhecer a maioria dos artigos tecnológicos dos principais blogs de notícia, compreender o significado dos artigos e quem são seus autores, entender sobre as tendências do mercado e, finalmente, conectar um indivíduo que tem uma história pra contar (empresa) com um jornalista que se interessa pelo tema.

As tecnologias necessárias para realizar todas essas funções são:

- *Crawler e parser* de blogs de notícias.
- Banco de dados de *Big Data clusterizado* de maneira flexível para poder realizar tarefas de *MapReduce* de maneira escalável.
- *Back-end* em Python.
- *Renderização de templates* no *front-end* utilizando React.js
- Inteligência artificial capaz de realizar análise de sentimentos e *keywords* nos artigos.

Uma das características mais interessantes do Cogentio é a necessidade do usuário poder acessar um banco de dados imenso em instantes. Esse requisito não funcional de tempo faz com que a imensa tarefa de buscar informação tenha de ser dividida em diversas máquinas, organizadas em um cluster.

Para realizar tal feito é utilizado um modelo de programação chamado MapReduce, que consiste em mapear a grande tarefa em menores funções e reduzir o resultado das sub funções em um *output* único.

2.2.2: Segundo Caso de Estudo – Camiolog

A empresa Camiolog foi fundada em 2011 pelo ex vice presidente da Google, Carter Maslan e outros engenheiros de software, em San Mateo, CA. A empresa desenvolve um sistema integrado de

monitoramento de imagens utilizando smartphones (iOS e Android) como câmeras e também como ferramenta de visualização dos dados. A estrutura de software desenvolvida pela Cheesecake possui um sistema de redes neurais que se treina utilizando as imagens das câmeras (em conjunto com outputs de algoritmos de detecção de movimento) como inputs da rede. Para definir o peso dos neurônios, durante o treinamento, a equação utiliza o input que o usuário insere no programa através de ações positivas (*Star*) e ações negativas (*Hide*). Se adaptando ao interesse do usuário, o sistema organiza os eventos relevantes gravados pelas câmeras em segmentos e é capaz de resumir dias de gravação em minutos de informações relevantes.

Para que seja possível manter servidores integrados com milhares de câmeras é essencial distribuir o processamento de movimento nos dispositivos que realizam o *upload* das imagens. Sendo assim, os dispositivos móveis – utilizando OpenCV – tratam as imagens em tempo real e apenas enviam para os servidores os conteúdos que contêm movimento. Dessa maneira, torna-se escalável processar o conteúdo de imagens em tempo real de milhares (ou até milhões) de câmeras.

Para que o sistema funcione de maneira segura, é essencial que o aplicativo da câmera seja extremamente robusto em cenários nos quais seu funcionamento se dá de maneira ininterrupta por meses, ou até anos. Essa tarefa se torna mais complicada devido à segmentação da linguagem Android – atualmente possui versões de diversas empresas que o utilizam em seus produtos. Em conjunto com a segmentação, os algoritmos do OpenCV de detecção de movimento podem consumir a memória do dispositivo, desligando a câmera e criando um cenário crítico para o usuário.

Os desafios gerenciais encontrados na Camiolog tem um caráter mais técnico (quando comparados aos do primeiro caso de estudo) devido aos riscos – em tempo real – de seu produto falhar. Essa característica salienta a necessidade de aprimoramento na gestão dos processos de desenvolvimento software na Cheesecake Labs, para garantir sempre o perfeito funcionamento de *features* de risco no produto realizado com a Camiolog.

Capítulo 3: Normas, Metodologias e Modelos

Neste capítulo serão estudadas as normas, metodologias e modelos mais consagrados de gestão projetos, gestão de desenvolvimento de software e modelagem de processos. A metodologia de escolha das referencias a serem estudadas foi buscar na web quais eram os temas que possuíam mais citações (que mais estão em pauta).

Na seção de gestão de projetos serão estudados o Guia PMBoK e a norma 21500; Na seção de gestão de desenvolvimento de software serão estudados o CMMI-DEV e o MPS.br; Na seção de modelagem de processos serão estudados o BPMN, o UML e o IDEF-0.

3.1: Gestão de Projetos

Nesta seção será feita uma breve introdução e resumo das principais normas, metodologias e modelos de gestão de projetos.

3.1.1: Project Management Body of Knowledge (PMBoK)

3.1.1.1: Histórico

O Guia PMBoK foi publicado pela primeira vez pelo Project Management Institute (PMI) como um *white paper* em 1983 na tentativa de documentar e padronizar as práticas que são normalmente aceitas na gestão de projetos. A primeira edição foi publicada em 1996, seguida pelas segunda e terceira edições em 2000 e 2004.

A última versão do Guia PMBOK é a quinta edição que foi publicada em 2013 [1] em Inglês, Árabe, Chinês, Francês, Alemão, Italiano, Japonês, Coreano, Português, Russo e Espanhol.

3.1.1.2: Definição

O Guia PMBOK identifica subconjuntos de conhecimentos utilizados em gerenciamento de projetos que são amplamente reconhecidos como boas práticas. Uma boa prática não significa que o conhecimento deve ser aplicado uniformemente a todos os projetos, sem considerar se são ou não apropriados.

O Guia PMBOK também fornece e promove um vocabulário comum para se discutir, escrever e aplicar o gerenciamento de projetos - possibilitando o intercâmbio eficiente de informações entre os profissionais de gestão de projetos.

O guia é baseado em processos e subprocessos para descrever de forma organizada o trabalho a ser realizado durante o projeto. Os processos descritos se relacionam e interagem durante a condução do trabalho e a descrição de cada um deles pode ser feita em termos de:

- Entradas (documentos, planos, desenhos etc.);
- Ferramentas e técnicas (que se aplicam às entradas);
- Saídas (documentos, produtos etc.)

3.1.1.3: Ciclo de Vida e Organização de um Projeto

O Guia PMBOK reconhece 42 processos que recaem em 5 grupos de processos e 9 áreas de conhecimento.

Os cinco grupos de processo definidos pela quinta edição do Guia PMBOK:

- Iniciação
- Planejamento
- Execução
- Monitoramento e Controle
- Encerramento

As dez áreas de conhecimento definidas pela quinta edição do Guia PMBOK:

- Integração
- Escopo
- Tempo
- Custo
- Qualidade
- Recursos Humanos
- Comunicações
- Riscos
- Aquisições
- Partes Interessadas

A divisão dos 42 processos em seus respectivos grupos e áreas de interesse pode ser observada na figura 2.

Knowledge Areas	Project Management Process Groups				
	Initiating Process Group	Planning Process Group	Executing Process Group	Monitoring & Controlling Process Group	Closing Process Group
4. Project Integration Management	4.1 Develop Project Charter	4.2 Develop Project Management Plan	4.3 Direct and Manage Project Execution	4.4 Monitor and Control Project Work 4.5 Perform Integrated Change Control	4.6 Close Project or Phase
5. Project Scope Management		5.1 Collect Requirements 5.2 Define Scope 5.3 Create WBS		5.4 Verify Scope 5.5 Control Scope	
6. Project Time Management		6.1 Define Activities 6.2 Sequence Activities 6.3 Estimate Activity Resources 6.4 Estimate Activity Durations 6.5 Develop Schedule		6.6 Control Schedule	
7. Project Cost Management		7.1 Estimate Costs 7.2 Determine Budget		7.3 Control Costs	
8. Project Quality Management		8.1 Plan Quality	8.2 Perform Quality Assurance	8.3 Perform Quality Control	
9. Project Human Resource Management		9.1 Develop Human Resource Plan	9.2 Acquire Project Team 9.3 Develop Project Team 9.4 Manage Project Team		
10. Project Communications Management	10.1 Identify Stakeholders	10.2 Plan Communications	10.3 Distribute Information 10.4 Manage Stakeholder Expectations	10.5 Report Performance	
11. Project Risk Management		11.1 Plan Risk Management 11.2 Identify Risks 11.3 Perform Qualitative Risk Analysis 11.4 Perform Quantitative Risk Analysis 11.5 Plan Risk Responses		11.6 Monitor and Control Risks	
12. Project Procurement Management		12.1 Plan Procurements	12.2 Conduct Procurements	12.3 Administer Procurements	12.4 Close Procurements

Figura 2 – Divisão dos processos da PMBoK em seus diferentes grupos e áreas de conhecimento. Fonte: [1]

3.1.1.4: Áreas de Conhecimento Estudadas

Nesta seção será feita uma breve introdução e resumo das áreas de conhecimentos utilizadas para modelagem e aprimoramento de processos no Capítulo 4. Este documento não abrange todas as áreas de conhecimento para simplificar o estudo do problema. Os conceitos aqui descritos usam como referência os manuais da PMI – Project Management Institute [2].

3.1.1.4.1: Gerenciamento da Integração do Projeto

O gerenciamento da integração do projeto é o núcleo do gerenciamento de projetos, sendo composta dos processos do dia-a-dia com os quais o gerente de projetos conta para garantir que todas as partes funcionem juntas. Essa é única área que está presente em todos os grupos de processos. É essencial para garantir que o projeto prossiga integrado do início ao fim.

Gerenciar a integração do projeto é garantir que os componentes do projeto precisam trabalhar juntos – e é papel do gerente de projetos fazer com que isso aconteça. Exige habilidades em negociação e gerenciamento de conflitos de interesses. Também exige habilidades gerais de gerenciamento, boa comunicação, organização, familiaridade técnica com o produto, etc.

A Gestão da integração do projeto pode ser dividida em três partes: o desenvolvimento do plano do projeto, a execução do plano do projeto e o controle de mudanças no projeto.

3.1.1.4.2: Gestão do Escopo do Projeto

O gerenciamento do escopo, ou âmbito, do projeto inclui a definição do trabalho necessário para sua conclusão e serve como guia, ou ponto de referência, para determinar quais tarefas não estão incluídas ou não são necessárias.

O escopo é o foco do projeto. O escopo do projeto difere-se do escopo do produto na medida em que o escopo do projeto define o trabalho necessário para fazer o produto, e o escopo do produto define os recursos (atributos e comportamentos) do produto que está sendo criado.

A maioria dos projetos passam por um processo para determinar seu custo e valor. Eles são selecionados com base em diversas condições: oportunidade, necessidade, exigências do cliente, mudança de legislações, entre outros. O escopo/âmbito do projeto é essencial para entender corretamente qual será a relação custo/benefício e determinar se o projeto vale a pena ser feito.

O planejamento do escopo do projeto é feito através do processo chamado plano de gerenciamento do escopo. Para isso, tanto o gerente quanto a equipe precisam ter uma visão unificada sobre quais são os componentes do projeto, dos seus requisitos, da expectativa dos stakeholders do projeto e como o projeto se encaixa em suas necessidades de negócio.

O resultado dos processos de planejamento de escopo é a declaração de escopo. A declaração de escopo diz o que está dentro e o que está fora do projeto, de maneira clara e sem ambiguidades. É importante que a declaração de escopo seja bem-feita e que haja acordo sobre ela. Quando a declaração de escopo estiver pronta, a equipe do projeto, os stakeholders, o patrocinador do projeto e o gerente de projetos não deverão mudar o escopo – a menos que haja um motivo muito forte que justifique essa mudança (que quase certamente implica impactos no custo do projeto).

3.1.1.4.3: Gestão de Tempo do Projeto

O objetivo da gestão do tempo de projeto é descrever os processos requeridos para o término do mesmo, garantindo que os prazos, definidos em um cronograma de atividades, sejam cumpridos. Os principais processos desta gestão são:

- Definições das Atividades: identificação das atividades específicas do cronograma que necessitam ser executadas para produzir os diversos tangíveis do projeto;
- Sequenciar Atividades: identificação e documentação das dependências entre as atividades do cronograma;
- Estimativa de Recursos de Atividade: estimativa do tipo e das quantidades dos recursos requeridos para executar cada atividade do cronograma;
- Estimativa de Duração de Atividade: estimativa do período que será necessário para conclusão individual de cada atividade do cronograma;
- Desenvolvimento do Cronograma: análise das sequências das atividades, suas dependências, durações e recursos requeridos para criar o cronograma;
- Controle do Cronograma: controle das alterações efetuadas no cronograma;

A gestão do tempo de projeto e a gestão do custo do projeto são as áreas de maior exigência, pois são as mais visíveis em sua gestão. Algumas das ferramentas clássicas de gestão de tempo de projeto são o PERT/CPM e o Diagrama de Gantt.

3.1.1.4.4: Gestão de Custos do Projeto

A gestão do custo do projeto agrega os processos que envolvem planejamento, estimativa, orçamento e controle de custos, que serão necessários para a conclusão do projeto a partir de uma previsão orçamentária.

Os processos de gestão do custo do projeto incluem:

- Estimativa de Custo: desenvolver uma aproximação dos gastos com os recursos necessários para execução do projeto;
- Orçamento de Custo: agregar os custos estimados de atividades ou de pacotes individuais de trabalho para estabelecer uma base de custo;
- Controle de Custo: influenciar nos fatores que geram uma variação de custo e controlar as mudanças de orçamento do projeto;

A gestão de custos é um processo em que se utiliza um conjunto de técnicas multidisciplinares, que permite compreender a origem dos custos. Esse processo pode conduzir ao

aumento de proveitos, reduções de custos e obtenção de melhores níveis de produtividade. Surgem assim várias metodologias na gestão de custos: Modelo ABC, Custo alvo, Just in time, Qualidade, Teoria das Restrições, Gestão do Tempo, entre outras.

3.1.1.4.5: Gestão da Qualidade do Projeto

Qualidade do projeto é definida como “o grau até o qual um conjunto de características inerentes satisfaz as necessidades”.

Segundo o PMI, “um projeto com qualidade é aquele concluído em conformidade com os requisitos, especificações e adequação ao uso”.

Os principais processos da gestão da qualidade do projeto são:

- Planejamento da qualidade: identificação dos padrões de qualidade relevantes para o projeto e determinação de como atender a esses padrões;
- Realizar a garantia da qualidade: aplicação das atividades de qualidade planejadas e sistemáticas para assegurar que o projeto empregará todos os processos necessários para atender os requisitos;
- Realizar o controle de qualidade: monitoramento dos resultados específicos do projeto, a fim de determinar se os mesmos estão de acordo com os padrões relevantes de qualidade, e identificação de maneiras para eliminar as causas de um desempenho insatisfatório.

3.1.2: ISO 21500

3.1.2.1: Histórico

Em 1997 a ISO, maior organismo de normalização mundial, publicou o ISO 10006 para abordar a temática de sistemas de gestão de qualidade. O ISO 10006 não ganhou popularidade como a norma da série ISO 9000 ou como os outros principais padrões de gerenciamento de projetos do mundo, como PMBOK ou prince2. Alguns países membros da ISO criaram normas razoavelmente populares de Gestão de Projetos, como o BSI 6079 [3].

A iniciativa ISO 21500 foi iniciada em 2006 pelo British Standard Institute, uma organização membro da ISO. Foram 31 países envolvidos neste trabalho e 5 observando seu desenvolvimento. O presidente do grupo, Dr. Jim Gordon, representava o Reino Unido e todo o secretariado foi hospedado pela ANSI (instituto que adotou o PMBOK como padrão nacional para gerenciamento de projetos em 1999). A versão final da ISO 21500 foi publicada em setembro de 2012.

3.1.2.2: Definição

A ISO 21500 ajuda na transferência de conhecimentos entre projetos e organizações, resultando na melhoria das entregas dos projetos. Facilita processos de concorrência mais eficientes, especialmente em grandes projetos internacionais através do uso de terminologia de gerenciamento de projeto consistente. Permite que as organizações multinacionais coordenem seus sistemas e processos de gerenciamento de projeto. Facilita a mobilidade do pessoal de gestão de projeto e a sua capacidade para trabalhar em projetos internacionais. Fornecem um quadro de princípios genéricos de gerenciamento de projeto e processos que poderiam ser desenvolvidos para o avanço da profissão de gerenciamento de projeto e das organizações.

Assim como o PMBoK, descrito na seção 3.1.1, a ISO 21500 divide os processos em cinco grupos:

- Iniciação
- Planejamento
- Execução
- Controle
- Encerramento

Diferentemente do PMBoK, descrito na seção 3.1.1, os processos são organizados em função de assuntos (*subjects*). Os dez assuntos definidos pela ISO 21500 são:

- Integração
- Escopo
- Tempo
- Custo
- Qualidade
- Recursos
- Comunicações
- Riscos
- Aquisições
- Partes Interessadas

A figura 3 mostra a inter-relação entre os processos e assuntos da ISO 21500.

Subject groups	Process groups				
	Initiating	Planning	Implementing	Controlling	Closing
Integration	4.3.2 Develop project charter	4.3.3 Develop project plans	4.3.4 Direct project work	4.3.5 Control project work 4.3.6 Control changes	4.3.7 Close project phase or project 4.3.8 Collect lessons learned
Stakeholder	4.3.9 Identify stakeholders		4.3.10 Manage stakeholders		
Scope		4.3.11 Define scope 4.3.12 Create work breakdown structure 4.3.13 Define activities		4.3.14 Control scope	
Resource	4.3.15 Establish project team	4.3.16 Estimate resources 4.3.17 Define project organization	4.3.18 Develop project team	4.3.19 Control resources 4.3.20 Manage project team	
Time		4.3.21 Sequence activities 4.3.22 Estimate activity durations 4.3.23 Develop schedule		4.3.24 Control schedule	
Cost		4.3.25 Estimate costs 4.3.26 Develop budget		4.3.27 Control costs	
Risk		4.3.28 Identify risks 4.3.29 Assess risks	4.3.30 Treat risks	4.3.31 Control risks	
Quality		4.3.32 Plan quality	4.3.33 Perform quality assurance	4.3.34 Perform quality control	
Procurement		4.3.35 Plan procurements	4.3.36 Select suppliers	4.3.37 Administer procurements	
Communication		4.3.38 Plan communications	4.3.39 Distribute information	4.3.40 Manage communications	

Figura 3 – Tabela dos processos da ISO 21500 em função dos grupos de processos e grupos de assunto. Fonte: [4]

3.1.3: Comparação entre PMBOK e ISO 21500

Esse capítulo traz uma breve comparação entre PMBoK e ISO 21500 [5, 6 e 7].

3.1.3.1: Grupos de Processos

Com relação aos grupos de processos as diferenças são mínimas, apenas mudanças de nomenclatura foram feitas. A tabela 1 mostra tal relação.

Tabela 1 – Relação entre os grupos de processos da ISO 21500 e do PMBoK. Fonte: [5]

Project Management Process Groups	
ISO 21500	PMBoK® Guide
Initiating	Initiating
Planning	Planning
Implementing	Executing
Controlling	Monitoring and Controlling
Closing	Closing

3.1.3.2: Grupos de Assuntos e Áreas de Conhecimento

As diferenças entre as áreas de conhecimento estudadas, citadas no capítulo 3.1.1.4, e seus respectivos grupos de assunto são apresentadas na tabela 2.

Tabela 2 – Relação entre os áreas de conhecimento da ISO 21500 e do PMBoK. Fonte: [5]

ISO 21500 - Subjects	PMBoK® Guide – Knowledge Areas
Integration	Integration
Stakeholder	X
Scope	Scope
Resource	Human Resources
Time	Time
Cost	Cost
Risk	Risk
Quality	Quality
Procurement	Procurement
Communication	Communication

3.1.3.2.1: Integração

Os aspectos de integração são praticamente similares, adicionando o processo "coletar lições aprendidas", focado em gestão do conhecimento do projeto, para a ISO 21500 é um diferencial em um contexto em que mais e mais profissionais e metodologias dizem que o conhecimento é o mais importante recurso do projeto e, portanto, merece ser tratado como disciplina separada na disciplina de gestão de projeto (Tabela 3).

Tabela 3 – Comparação entre ISO 21500 e PMBoK para integração. Fonte: [5]

ISO 21500	PMBoK® Guide
4.3.2 Develop Project Charter	4.1 Develop Project Charter
4.3.3 Develop Project Plans	4.2 Develop Project Management Plan
4.3.4 Direct Project Work	4.3 Direct and Manage Project Execution
4.3.5 Control Project Work	4.4 Monitor and Control Project Work
4.3.6 Control Changes	4.5 Perform Integrated Change Control
4.3.7 Close Project Phase or Project	4.6 Close Project or Phase
4.3.8 Collect Lessons Learned	

No PMBoK existe um plano de gerenciamento de projeto que consolida e integraliza todos os documentos, já a ISO 21500 requer o desenvolvimento de três tipos de planos:

- O Plano do Projeto: descreve o que deve ser alcançado pelo projeto em disciplinas separadas, como escopo, tempo, custo e outros.
- O Plano de Gerenciamento do Projeto: descreve os processos de gerenciamento de projetos.
- Os Planos Auxiliares: utilizado para complementar os outros documentos.

3.1.3.2.2: Escopo

Não há nenhum processo como Verificar o Escopo (Verify Scope) na norma ISO 21500, sendo que nenhum processo ISO 21500 produz uma saída de Entregas aceitas (Accepted deliverables), que é a saída mais importante do processo Verificar o Escopo, da PMBoK.

Outra pequena alteração é o fato da ISO 21500 possuir o processo Definir Atividades (Define Activities) no assunto de Escopo, pois o PMBoK possui esse processo na área de Tempo. A Tabela 4 mostra a comparação geral.

Tabela 4 – Comparação entre ISO 21500 e PMBoK para Escopo. Fonte: [5]

ISO 21500	PMBok® Guide
4.3.11 Define Scope	5.1 Collect Requirements
	5.2 Define Scope
4.3.12 Create Work Breakdown Structure	5.3 Create WBS
4.3.13 Define Activities	6.1 Define Activities (taken from Time Management Knowledge Area)
	5.4 Verify Scope
4.3.14 Control Scope	5.5 Control Scope

3.1.3.2.3: Tempo

Dois processos que existem no PMBoK, Definir Atividades (Define Activities) e Estimar Recursos da Atividade (Estimate Activity Resources), não estão incluídos na norma ISO 21500. Na norma ISO 21500 os respectivos processos foram movidos para a seção de Escopo, discutida na seção anterior (Tabela 5)

Tabela 5 – Comparação entre ISO 21500 e PMBoK para Tempo. Fonte: [5]

ISO 21500	PMBok® Guide
Moved to Scope subject	6.1 Define Activities
4.3.21 Sequence Activities	6.2 Sequence Activities
Moved to Scope subject	6.3 Estimate Activity Resources
4.3.22 Estimate Activity Durations	6.4 Estimate Activity Durations
4.3.23 Develop Schedule	6.5 Develop Schedule
4.3.24 Control Schedule	6.6 Control Schedule

3.1.3.2.4: Custos

Não existem mudanças relevantes na seção de custos (Tabela 6)

Tabela 6 – Comparação entre ISO 21500 e PMBoK para Custos. Fonte: [5]

ISO 21500	PMBok® Guide
4.3.25 Estimate Costs	7.1 Estimate Costs
4.3.26 Develop Budget	7.2 Determine Budget
4.3.27 Control Costs	7.3 Control Costs

3.1.3.2.5: Qualidade

Tambem não existem mudanças relevantes na seção de qualidade (Tabela 7).

Tabela 7 – Comparação entre ISO 21500 e PMBoK para Qualidade. Fonte: [5]

ISO 21500	PMBok® Guide
4.3.32 Plan Quality	8.1 Plan Quality
4.3.33 Perform Quality Assurance	8.2 Perform Quality Assurance
4.3.34 Perform Quality Control	8.3 Perform Quality Control

3.2: Gestão de Desenvolvimento Software

3.2.1: CMMI

3.2.1.1: Histórico

O Capability Maturity Model Integration - CMMI surgiu durante a década de 1980 como um modelo para avaliação de risco na contratação de empresas de software pelo Departamento de Defesa dos Estados Unidos. Para desenvolver esse processo, foi criado junto a Carnegie-Mellon University o SEI (Software Engineering Institute), o qual além de ser responsável pela evolução da família CMM, realiza diversas outras pesquisas em engenharia de software [8].

O CMMI não é um processo, mas sistema de melhoria de desempenho para as organizações. Baseia-se nos objetivos de desempenho de negócio de uma organização e fornece um conjunto de práticas para a melhoria dos processos, resultando em um sistema de melhoria de desempenho que abre o caminho para melhores operações e desempenho. Destaca-se em relação a outras abordagens, pois o CMMI não apenas auxilia a melhorar os processos organizacionais, mas também a maneira de melhorar a capacidade dos processos [9].

Entende-se por capacidade de um processo a habilidade com que este alcança o resultado desejado. Um modelo tem como objetivo estabelecer - com base em estudos, históricos e conhecimento operacional - um conjunto de "melhores práticas" que devem ser utilizadas para um fim

específico. O CMMI tem como origens em três outros modelos de maturidade - SW-CMM (*SEI Software CMM*), EIA SECM (*Electronic Industries Alliances's Systems Engineer Capability Model*) e IPD-CMM (*Integrated Product Development CMM*).

3.2.1.2: Definição

O CMMI (Capability Maturity Model - Integration ou Modelo de Maturidade em Capacitação - Integração) é um modelo de referência que contém práticas (Genéricas ou Específicas) necessárias para alcançar maturidade em disciplinas específicas: Systems Engineering (SE - Engenharia de Sistemas), Software Engineering (SW - Engenharia de Software), Integrated Product and Process Development (IPPD - Desenvolvimento Integrado de Processo e Produto) e Supplier Sourcing (SS).

O CMMI é uma evolução do CMM, que incorpora a integração, e procura estabelecer um modelo único para o processo de melhoria corporativo, integrando diferentes modelos e disciplinas. O CMMI foi construído considerando três dimensões principais: pessoas, ferramentas e procedimentos. O processo serve para unir essas dimensões.

As melhores práticas do CMMI são publicados em documentos chamados modelos, cada um dos quais aborda uma área diferente de seu interesse. A versão atual, CMMI versão 1.3, fornece modelos para três áreas de interesse:

- **CMMI para o Desenvolvimento (CMMI-DEV):** foi lançado em novembro de 2010 e aborda os processos de desenvolvimento de produtos e serviços.
- **CMMI para Aquisição (CMMI-ACQ):** foi lançado em novembro de 2010 e aborda a gestão da cadeia de suprimentos, aquisição e processos de terceirização no governo e na indústria.
- **CMMI para Serviços (CMMI-SVC):** foi lançado em novembro de 2010 e aborda a orientação para a prestação de serviços dentro de uma organização e para os clientes externos.

3.2.1.3: Representações

O CMMI possui duas representações: "contínua" ou "por estágios". Estas representações permitem à organização utilizar diferentes caminhos para a melhoria de acordo com seu interesse:

Representação Contínua: possibilita à organização utilizar a ordem de melhoria que melhor atende os objetivos de negócio da empresa. É caracterizado por: Níveis de Capacidade (Capability Levels):

Nível 0: Incompleto (Ad-hoc)

Nível 1: Executado

Nível 2: Gerenciado / Gerido

Nível 3: Definido

Nível 4: Gerenciado quantitativamente (removido da versão 1.3).

Nível 5: Em otimização (removido da versão 1.3).

Nesta representação a capacidade é medida por processos separadamente, onde é possível ter um processo com nível um e outro processo com nível cinco, variando de acordo com os interesses da empresa.

A representação contínua é indicada quando a empresa deseja tornar apenas alguns processos mais maduros, quando já utiliza algum modelo de maturidade contínua ou quando não pretende usar a maturidade alcançada como modelo de comparação com outras empresas.

Representação Por Estágios: disponibiliza uma sequência pré-determinada para melhoria baseada em estágios que não deve ser desconsiderada, pois cada estágio serve de base para o próximo. É caracterizado por Níveis de Maturidade (Maturity Levels):

Nível 1: Inicial (Ad-hoc)

Nível 2: Gerenciado / Gerido

Nível 3: Definido

Nível 4: Quantitativamente gerenciado / Gerido quantitativamente

Nível 5: Em otimização

Nesta representação a maturidade é medida por um conjunto de processos. Assim é necessário que todos os processos atinjam nível de maturidade dois para que a empresa seja certificada com nível dois. Se quase todos os processos forem nível três, mas apenas um deles estiver no nível dois a empresa não irá conseguir obter o nível de maturidade três.

Esta representação é indicada quando a empresa já utiliza algum modelo de maturidade por estágios, quando deseja utilizar o nível de maturidade alcançado para comparação com outras empresas ou quando pretende usar o nível de conhecimento obtido por outros para sua área de atuação.

A relação entre os estágios das diferentes representações podem ser observados na tabela 7.

Tabela 7 – Relação entre os estágios dos diferentes tipos de representação do CMMI. Fonte:

[10]

<i>Level</i>	<i>Continuous Representation Capability Levels</i>	<i>Staged Representation Maturity Levels</i>
Level 0	Incomplete	
Level 1	Performed	Initial
Level 2	Managed	Managed
Level 3	Defined	Defined
Level 4		Quantitatively Managed
Level 5		Optimizing

3.2.1.4: Áreas de Processos

Nesta seção explica-se, de maneira breve, do que são compostas as diversas áreas de processos definidas pelo CMMI.

Nível 1: Inicial (Ad-hoc)

Não possui áreas de processo.

Nível 2: Gerenciado / Gerido

- Gerenciamento de Requisitos - REQM (Requirements Management)
- Planejamento de Projeto - PP (Project Planning)
- Acompanhamento e Controle de Projeto - PMC (Project Monitoring and Control)
- Gerenciamento de Acordo com Fornecedor - SAM (Supplier Agreement Management)
- Medição e Análise - MA (Measurement and Analysis)
- Garantia da Qualidade de Processo e Produto - PPQA (Process and Product Quality Assurance)
- Gestão de Configuração - CM (Configuration Management)

Nível 3: Definido

- Desenvolvimento de Requisitos - RD (Requirements Development)
- Solução Técnica - TS (Technical Solution)
- Integração de Produto - PI (Product Integration)
- Verificação - VER (Verification)
- Validação - VAL (Validation)
- Foco de Processo Organizacional - OPF (Organizational Process Focus)

- Definição de Processo Organizacional - OPD (Organizational Process Definition)
- Treinamento Organizacional - OT (Organizational Training)
- Gerenciamento Integrado de Projeto - IPM (Integrated Project Management)
- Gerenciamento de Riscos - RSKM (Risk Management)
- Análise de Decisão e Resolução - DAR (Decision Analysis and Resolution)

Nível 4: Quantitativamente gerenciado / Gerido quantitativamente

- Desempenho de Processo Organizacional - OPP (Organizational Process Performance)
- Gerenciamento Quantitativo de Projeto - QPM (Quantitative Project Management)

Nível 5: Em otimização

- Gestão de Processo Organizacional - OPM (Organizational Process Management)
- Análise Causal e Resolução - CARs (Causal Analysis and Resolution)

Pode-se observar a divisão dos processos nos seus respectivos níveis de maturidade na figura 4.

Configuration Management	CM	2	Target Profile 2
Measurement and Analysis	MA	2	
Project Monitoring and Control	PMC	2	
Project Planning	PP	2	
Process and Product Quality Assurance	PPQA	2	
Requirements Management	REQM	2	
Supplier Agreement Management	SAM	2	
Decision Analysis and Resolution	DAR	3	Target Profile 3
Integrated Project Management	IPM	3	
Organizational Process Definition	OPD	3	
Organizational Process Focus	OPF	3	
Organizational Training	OT	3	
Product Integration	PI	3	
Requirements Development	RD	3	
Risk Management	RSKM	3	
Technical Solution	TS	3	
Validation	VAL	3	
Verification	VER	3	
Organizational Process Performance	OPP	4	Target Profile 4
Quantitative Project Management	QPM	4	
Causal Analysis and Resolution	CAR	5	Target Profile 5
Organizational Performance Management	OPM	5	

Figura 4 - Divisão dos processos em seus níveis de maturidade, definidos pela CMMI. Fonte: [10]

3.2.2: MPS.BR

O modelo Melhoria de Processo de Software Brasileiro (MPS.Br) é um projeto que começou em 2003 coordenado pela Associação para Promoção da Excelência do Software Brasileiro (SOFTEX) com o apoio Ministério da Ciência e Tecnologia (MCT), Financiadora de Estudos e Projetos (FINEP) e o Banco Interamericano de Desenvolvimento (BID). O MPS.Br é baseado nas normas ISO/IEC 12207 e ISO/IEC 15504. Essas normas são as mesmas em que o CMMI foi baseado, e é por isso que se pode dizer que os dois modelos tem equivalência.

Ambos os modelos possuem níveis de maturidade que definem a capacidade da empresa em trabalhar em projetos grandes e complexos. O CMMI varia do 1 ao 5 e o MPS.Br varia do G ao A, sendo que ao contrário do CMMI, o primeiro nível já exige que a empresa tenha determinados processos definidos. A escala de níveis pode ser expressa da seguinte forma:

G: Parcialmente Gerenciado

- F:** Gerenciado
- E:** Parcialmente Definido
- D:** Largamente Definido
- C:** Definido
- B:** Gerenciado Quantitativamente
- A:** Em Otimização

Os níveis do MPS.Br também são compostos por Áreas de Processos, que são os tópicos mais importantes para um processo de desenvolvimento de software e através deles é possível criar a uma tabela de equivalência dos níveis do CMMI e do MPS.Br [11], como observado na tabela 8:

Tabela 8 – Comparação entre os níveis de maturidade do CMMI e MPS.Br. Fonte: [11]

Comparação dos Níveis de Maturidade	
CMMI	MPS.Br
1	Não é definido
	G
2	F
	E
	D
3	C
4	B
5	A

Analisando a tabela, verifica-se que os níveis do MPS.Br permitem que a empresa implante processos de uma forma mais gradual. Essa ideia refletida para o mercado brasileiro de software permite que empresas de pequeno porte, que não possuem muito dinheiro para investir em metodologias e processos, possam tomar a iniciativa de defini-los.

Hoje, o MPS.Br e o CMMI em termos de qualidade de software possuem níveis equivalentes, mas com a vantagem de ser muito mais barato e há financiamento do BID para grupos de empresas que desejam se certificar.

3.3: Modelagem

3.3.1: Business Process Modeling Notation (BPMN)

O BPMN é um padrão para modelagem e fornece uma notação gráfica para a especificação de processos de negócios em um Business Process Diagram (BPD), ou Diagrama de Processos de Negócio, baseado em uma técnica de fluxograma muito semelhante ao de diagramas de atividades da Unified Modeling Language (UML). O BPMN é capaz de apoiar a gestão de processos de negócios tanto para usuários técnicos e usuários de negócios, fornecendo uma notação que é intuitiva para os usuários corporativos e ainda capaz de representar a semântica complexa do processo. A especificação BPMN também fornece um mapeamento entre os gráficos da notação para as construções subjacentes de linguagens de execução, particularmente a Business Process Execution Language [12].

O principal objetivo do BPMN é fornecer uma notação padrão que seja facilmente compreensível por todos os intervenientes do negócio. Estas partes interessadas incluem os analistas de negócios que criam e refinam os processos, os desenvolvedores técnicos responsáveis pela implementação dos processos e os gerentes de negócios que monitoram e gerenciam os processos. Consequentemente, o BPMN é destinado a servir como linguagem comum para fazer a ponte de comunicação que ocorre com frequência entre o design de processos de negócios e implementação.

3.3.2: Unified Modeling Language (UML)

Os esforços para a criação da UML tiveram início em outubro de 1994, quando Rumbaugh se juntou a Booch na Rational Software [13]. Com o objetivo de unificar os métodos Booch e OMT, decorrido um ano de trabalho, foi lançado, em outubro de 1995, o esboço da versão 0.8 do Unified Process - Processo Unificado (como era conhecido). Nesta mesma época, Jacobson se associou à Rational Software e o escopo do projeto da UML foi expandido para incorporar o método OOSE. Nasceu então, em junho de 1996, a versão 0.9 da UML.

Finalmente em 1997, a UML foi aprovada como padrão pelo OMG (Object Management Group), um consórcio internacional de empresas que define e ratifica padrões na área de Orientação a Objetos.

UML 2.2, conforme a OMG, possui 14 tipos de diagramas, divididos em duas grandes categorias: Estruturais e Comportamentais. Sete tipos de diagramas representam informações estruturais, e os outros sete representam tipos gerais de comportamento, incluindo quatro em uma

subcategorias que representam diferentes aspectos de interação. Estes diagramas podem ser visualizados de forma hierárquica, como apresentado na figura 5:

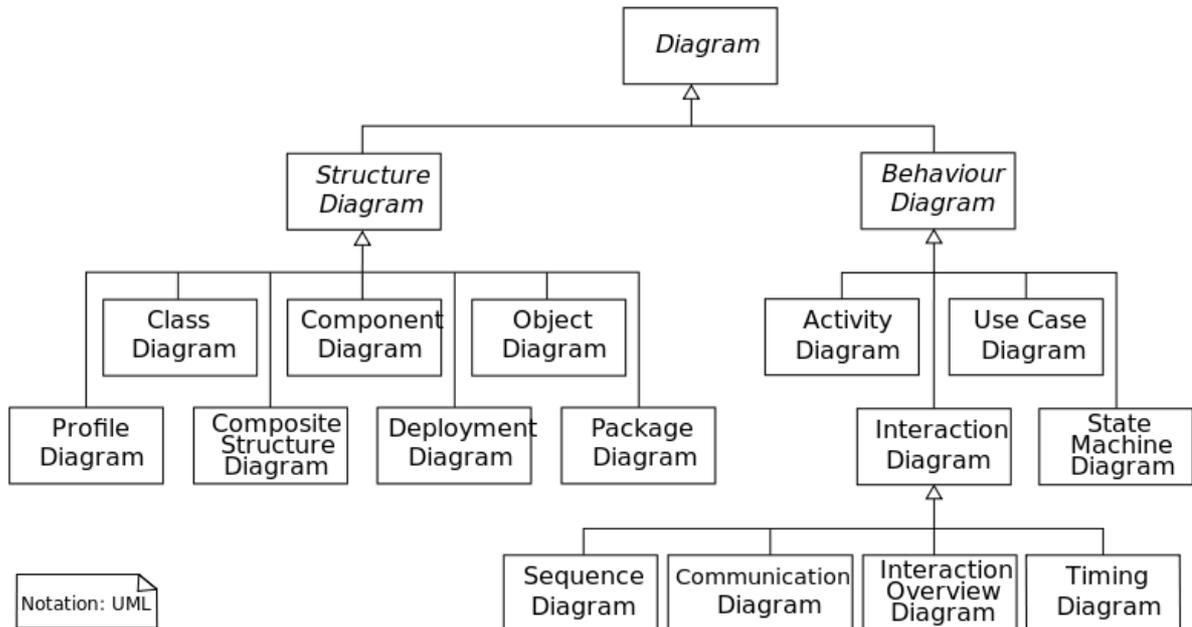


Figura 5 – Diagrama dos diferentes tipos de diagramas UML. Fonte: [14]

3.3.3: IDEF-0

IDEF (Integration Definition for Function Modeling) é uma família integrada de métodos para modelagem baseada em representações de diagramas, incluindo uma larga variedade de técnicas. Todas estas técnicas estão formalizadas no FIPS (Federal Information Processing Standards). O IDEF0, que é o primeiro conjunto de padrões do IDEF, processa uma coleção de atividades e outras ações utilizando-se de ICOMs (Input Control Output Mechanism). O ICOM não inclui apenas dados e informações mas também tudo que pode ser descrito como sendo um processo (esquema, estimativa, regulamentos, produtos, etc). O ICOM é uma representação gráfica de uma tarefa ou um conjunto de tarefas, que possui "terminais" para que possa ser alimentada ou alimentar outras ICOMs. Esses "terminais" recebem o nome de entrada, controle, saídas e mecanismos. A entrada recebe o dado a ser convertido pela atividade, o controle agrega responsabilidade de como e quando a entrada deve ser processada e executada, a saída apresenta o resultado de como a entrada foi processada e o mecanismo representa quem deve executar esta atividade (pode ser uma pessoa, equipamento, máquina ou outras organizações).

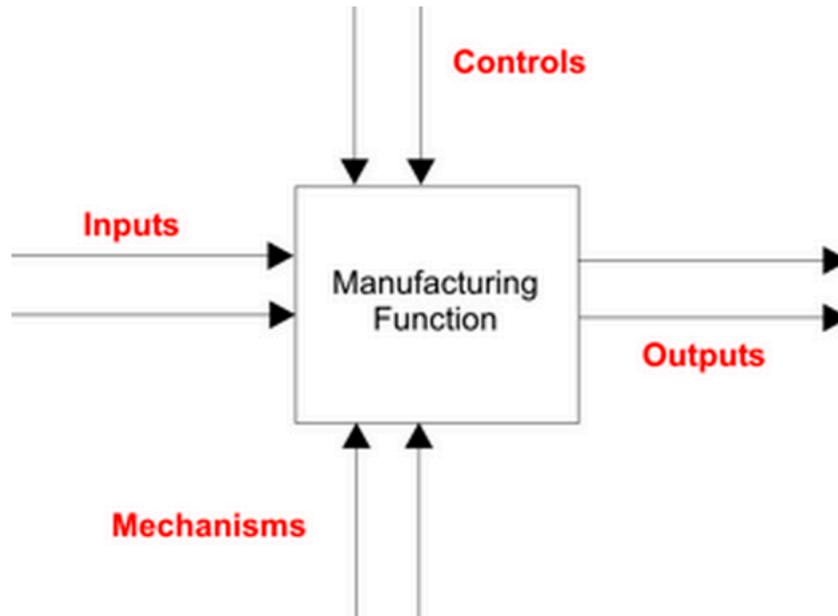


Figura 6 – Exemplo de ICOM. Fonte: [15]

O modelo funcional IDEF0 é composto por um conjunto de ICOMs, setas e caixas (Figura 6). Cada atividade ou função é conceitualmente representada por uma caixa retangular. Cada atividade pode ser decomposta em vários níveis. Estes sub-níveis seguem as mesmas convenções. Portanto um modelo completo de IDEF0 é uma representação hierárquica do processo composta por atividades ou funções em quantos níveis forem necessários.

Capítulo 4: Modelagem e Geração de Melhorias

Neste capítulo os processos da Cheesecake Labs são modelados na perspectiva de gestão de projetos e desenvolvimento de software. Para cada um dos casos, em primeira instancia são escolhidas as normas, metodologias e modelos que melhor satisfazem as restrições dos problemas; Em segunda instância, são criados modelos de como os processos atualmente são realizados; Em seguida, é explicitado como são modelados os processos previamente descritos pelas normas citadas no capítulo 3; Em quarta instância, são propostas, e aplicadas, mudanças nos processos para que se tornem mais compatíveis com os conceitos propostos pelas normas;

4.1: Gestão de Projetos

Nesta seção serão realizadas as etapas previamente descritas para os processos de gestão de projetos.

4.1.1: Escolha da Metodologia de Gestão e Norma de Modelagem

Para a metodologia foi escolhido o Guia PMBoK por ser mais difundido e estabelecido no cenário de negócios norte-americano. Tal decisão foi tomada devido a pequena diferença encontrada entre seus processos, com relação à ISO 21500, e por possuir muito mais conteúdo disponível. Outro fator que foi levado em consideração para realizar essa decisão foi o fato de os clientes da Cheesecake Labs estarem localizados nos EUA - local onde a PMBoK é adotada como a principal norma nacional, e pela ANSI, desde 1999.

Para a modelagem dos processos de gestão de projeto e desenvolvimento de software foi escolhido a BPMN devido ao fato de possuir uma gama maior de recursos e ferramentas para modelar de maneira mais precisa aspectos temporais e funcionais dos processos.

4.1.2: Modelagem do Estado Atual

Foram modelados os processos considerados críticos pelo corpo administrativo da empresa Cheesecake Labs. Os gerentes deixaram claro que os processos referentes aos períodos de inicialização e planejamento são os mais críticos, pois são nessas etapas onde ocorrem os maiores desentendimentos com os clientes, a falta de estrutura e ferramentas para melhor organização e, como consequência, o término mal planejado de projetos. Pode-se observar a definição do escopo do projeto (processos críticos) na figura 7.

Knowledge Areas	Project Management Process Groups				
	Initiating Process Group	Planning Process Group	Executing Process Group	Monitoring & Controlling Process Group	Closing Process Group
4. Project Integration Management	4.1 Develop Project Charter	4.2 Develop Project Management Plan	4.3 Direct and Manage Project Execution	4.4 Monitor and Control Project Work 4.5 Perform Integrated Change Control	4.6 Close Project or Phase
5. Project Scope Management		5.1 Collect Requirements 5.2 Define Scope 5.3 Create WBS		5.4 Verify Scope 5.5 Control Scope	
6. Project Time Management		6.1 Define Activities 6.2 Sequence Activities 6.3 Estimate Activity Resources 6.4 Estimate Activity Durations 6.5 Develop Schedule		6.6 Control Schedule	
7. Project Cost Management		7.1 Estimate Costs 7.2 Determine Budget		7.3 Control Costs	
8. Project Quality Management		8.1 Plan Quality	8.2 Perform Quality Assurance	8.3 Perform Quality Control	
9. Project Human Resource Management		9.1 Develop Human Resource Plan	9.2 Acquire Project Team 9.3 Develop Project Team 9.4 Manage Project Team		
10. Project Communications Management	10.1 Identify Stakeholders	10.2 Plan Communications	10.3 Distribute Information 10.4 Manage Stakeholder Expectations	10.5 Report Performance	
11. Project Risk Management		11.1 Plan Risk Management 11.2 Identify Risks 11.3 Perform Qualitative Risk Analysis 11.4 Perform Quantitative Risk Analysis 11.5 Plan Risk Responses		11.6 Monitor and Control Risks	
12. Project Procurement Management		12.1 Plan Procurements	12.2 Conduct Procurements	12.3 Administer Procurements	12.4 Close Procurements

Figura 7 – Definição do escopo de estudo (seção crítica dos processos). Fonte: adaptado de [1]

Para compreender, documentar e explicitar os processos vigentes, referentes a Inicialização e Planejamento, buscou-se uma ferramenta de modelagem em BPMN que satisfizesse as necessidades de recursos e possuísse baixo custo. As necessidades de recursos previamente citadas podem ser descritas nos seguintes itens:

- Capacidade de trabalhar em um arquivo de BPMN em equipe (de maneira distribuída).
- Interface intuitiva e completa.
- Capacidade de exportar o documento como imagem.

Após uma busca detalhada, a ferramenta que melhor satisfez a necessidade da Cheesecake Labs foi a LucidChart [16]. A ferramenta LucidChart possui interface *web* (que possibilita o trabalho distribuído), baixo custo, funcionalidades intuitivas e todas as notações e elementos gráficos do

BPMN. Ela é utilizada por empresas consolidadas como a Disney, Netflix e DropBox, além de instituições renomadas como a NASA, MIT, Harvard e Stanford University.

Utilizando a ferramenta LucidChart, modelou-se os processos de Inicialização e Planejamento na empresa Cheesecake Labs (figura 8).

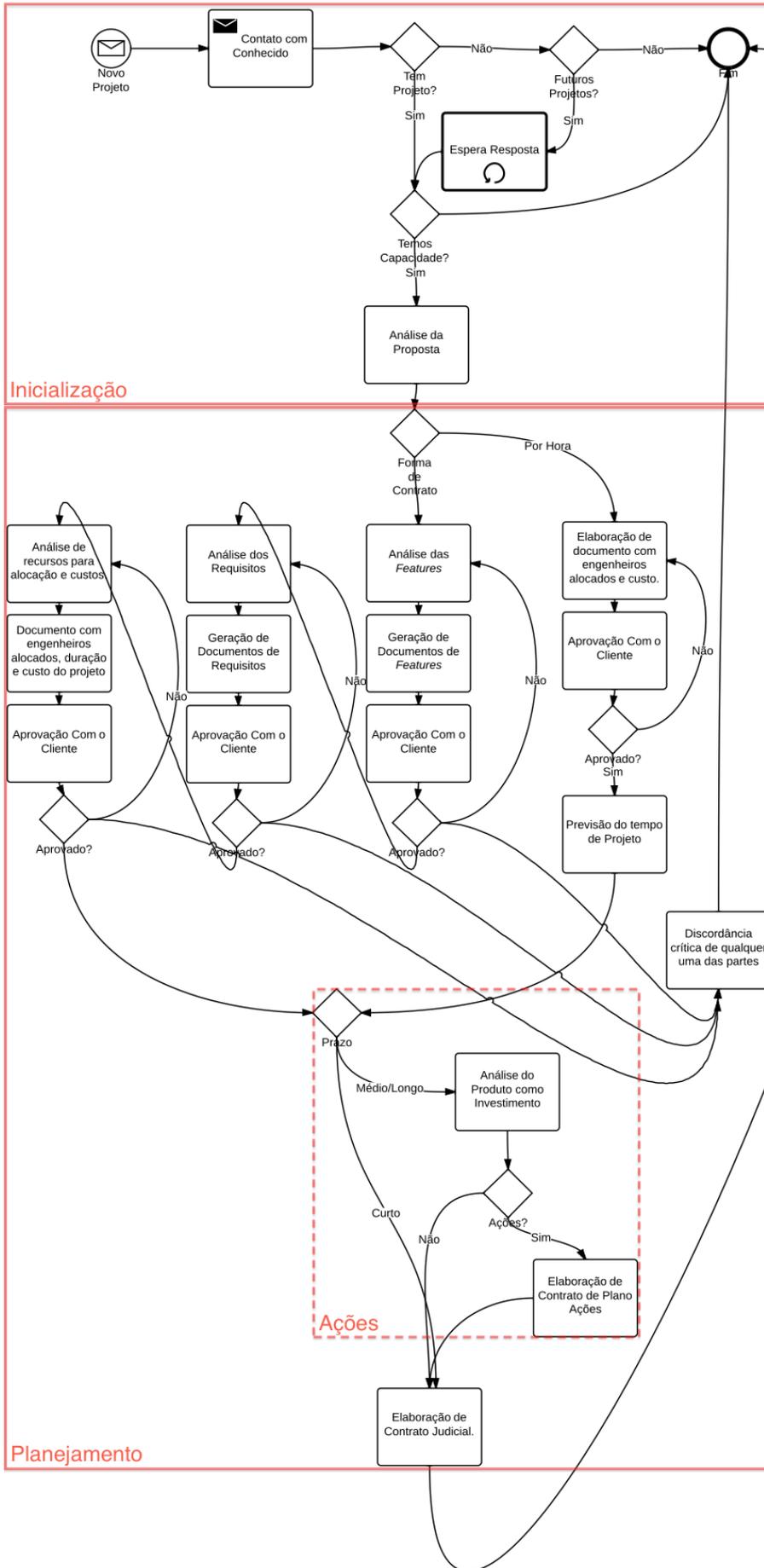


Figura 8 – Modelagem dos processos de Inicialização e Planejamento. Fonte: Autor

O modelo do estado atual dos processos pode ser dividido nas etapas que o compõem:

- **Inicialização**

O ponto de partida do processo é um evento inicial de "Novo Projeto", representando a iniciativa do corpo gestor da Cheesecake Labs de buscar uma nova oportunidade de negócios.

Em segunda instância, a atividade "Contato com Conhecido" representa os esforços dos vendedores de entrar em contato com entidades externas que possuem perfil de possíveis parceiros e clientes.

Em muitos casos os possíveis clientes não demonstram interesse inicialmente, mas a médio prazo retornam com novos projetos ou oportunidades com conhecidos. Para representar esse cenário assíncrono de vendas e espera dos vendedores, foi criada a atividade "Espera Resposta", em conjunto com suas portas lógicas.

Caso o contato com o cliente não tenha nenhum resultado o processo se finaliza no evento final "Fim".

Se a proposta de trabalho se concretizar, é essencial analisar se o escopo do projeto pertence ao nicho de mercado da Cheesecake Labs. Em muitos casos as propostas são de áreas muito distantes do escopo de trabalho da empresa. Essa verificação é representada no modelo pela porta lógica "Temos Capacidade".

Se a proposta pertencer ao nicho de mercado da Cheesecake Labs, são realizadas mais reuniões com os possíveis clientes para melhor compreender o caso de negócio (*business case*). Esse processo é representado no modelo pela atividade "Análise de Proposta".

- **Planejamento**

Recebendo como entrada o caso de negócio é essencial decidir se a forma de pagamento será realizada por horas trabalhadas ou por *features* de produto (que são partes funcionais do produto) realizadas. Em situações nas quais as especificações do produto podem sofrer muitas mudanças é essencial realizar o contrato por hora. Se os possíveis clientes possuírem especificações precisas e documentadas das *features* do produto, ou até de requisitos do projeto, faz mais sentido cobrar por *features*. O corpo gerencial da Cheesecake Labs explicita a importância da realização dessas tarefas de maneira precisa devido a experiências negativas com projetos baseado em *features* que mudavam constantemente de requisitos – devido ao caráter de *startup* das empresas. Essa decisão está representada no modelo pela porta lógica "Forma de Contrato". Seguem os dois caminhos que o projeto pode seguir:

- Por Hora

Se a decisão do corpo gerencial da Cheesecake for realizar o projeto cobrando por horas trabalhadas, é criado um documento explicitando como será feita a alocação de engenheiros e qual será o custo semanal. Esse processo é representado no modelo pela atividade "Elaboração de documento com engenheiros alocados e custo".

Em seguida, o documento previamente criado é apresentado aos clientes em busca de sua aprovação. Esse processo está representado no modelo pela atividade "Aprovação com o Cliente". Caso ele não seja aprovado a tarefa de criar o documento será repetida com o intuito de adequá-lo às necessidades da situação.

Se o documento for aprovado pelos clientes, é feita uma análise de qual é a expectativa de duração do projeto para futuras considerações. Esse processo é representado pela atividade "Previsão do Tempo do Projeto".

- Por *Feature*

Se a decisão do corpo gerencial da Cheesecake for realizar o projeto cobrando por *features* de produto, é feita uma análise do caso de negócio para melhor compreender quais serão as *features* a serem implementadas. Em muitos casos é essencial se reunir várias vezes com os clientes para identificar precisamente suas ideias e interesses. Essa processo está representado no modelo pela atividade "Análise de *Features*".

Após a análise das *features*, é criado um documento que as explicita detalhadamente para que seja possível conversar com os possíveis clientes sobre questões mais concretas. Esse processo está descrito pela atividade "Geração de Documento de *Features*".

O documento criado previamente é enviado para os potenciais clientes para aprovação. Esse processo é descrito pela atividade "Aprovação dos Clientes". Se os clientes não o aprovem é necessário realizar outra análise do problema e gerar outro documento a ser aprovado.

Em seguida é feita uma análise dos requisitos de projetos necessários para satisfazer as necessidades dos clientes, baseando-se no documento de *features* e em conversas – com os potenciais clientes – sobre como deve ser o desempenho do sistema em diferentes cenários. Esse processo é descrito pela atividade "Análise de Requisitos".

Após a análise, é criado um documento com os requisitos de projeto coletados. Esse processo está descrito pela atividade "Geração de Documento de Requisitos".

Em seguida, o documento de requisito passa pela aprovação dos clientes, como pode ser verificado na atividade "Aprovação com o Cliente".

Então, possuindo uma precisa definição de qual serão as *features* e requisitos necessários, é definido – em conjunto com o cliente – como será feita a alocação de recursos

humanos e qual deve ser o prazo do projeto. Nessa etapa também se definem quais serão os custos desse projeto para a Cheesecake e como será elaborado o orçamento para o possível cliente. Esse processo está descrito na atividade "Análise de recursos para alocação e custos".

Em seguida, são documentadas as conclusões retiradas da análise previamente realizada. Essa documentação é representada pela atividade "Geração de documento com engenheiros alocados, duração e orçamento do projeto".

Finalmente o documento previamente criado é enviado para aprovação com o cliente na atividade "Aprovação com o Cliente".

Em seguida, as duas vertentes – por *feature* e por hora – se juntam. Dependendo do prazo do projeto em questão pode ser feita uma análise do possível cliente como oportunidade de investimento através de planos de ações. Se o projeto for de médio ou longo prazo (2 – 4 anos) o caso de negócio será analisado, na perspectiva de investimento, no próximo processo. Essa divisão, em função da estimativa de duração, é representada pela porta lógica "Prazo".

Se o projeto for de médio ou longo prazo é feita a, previamente citada, análise de possível investimento juntamente com conselheiros norte americanos da Cheesecake. Nessa análise é levado em consideração não só o produto, mas o corpo de colaboradores, desenvolvedores, investidores e também se a empresa tem interesse nesse tipo de contrato ou não. Esse processo é definido pela atividade "Análise do Produto como Investimento".

Se o projeto for considerado viável para investimento, é elaborado um contrato de plano de ações, descrito na atividade "Elaboração de Contrato de Plano de Ações".

Finalmente, é elaborado e assinado um contrato judicial que leva em conta os documentos previamente citados. Esse processo representa o final da parte de planejamentos e está descrito na atividade "Elaboração de Contrato Judicial".

4.1.3: Modelagem do Estado Ideal

Nesta seção é explicitado como os processos de Inicialização e Planejamento são modelados através do Guia PMBoK. Devido a complexidade do modelo completo, os processos serão estudados separadamente (com entradas, ferramentas e saídas) para que o fluxo de tarefas entre as gerências se mantenha claro.

Utilizando a ferramenta LucidChart, foi feito um diagrama UML sequencial para ilustrar como seria o fluxo de processos no estado ideal (pautado pela norma PMBoK). Não foi possível modelar o sistema utilizando BPMN (assim como na seção de modelagem do estado atual) devido à complexidade dos diversos estados do sistema e suas muitas entradas e saídas. O diagrama do modelo ideal pode ser observado na Figura 9.

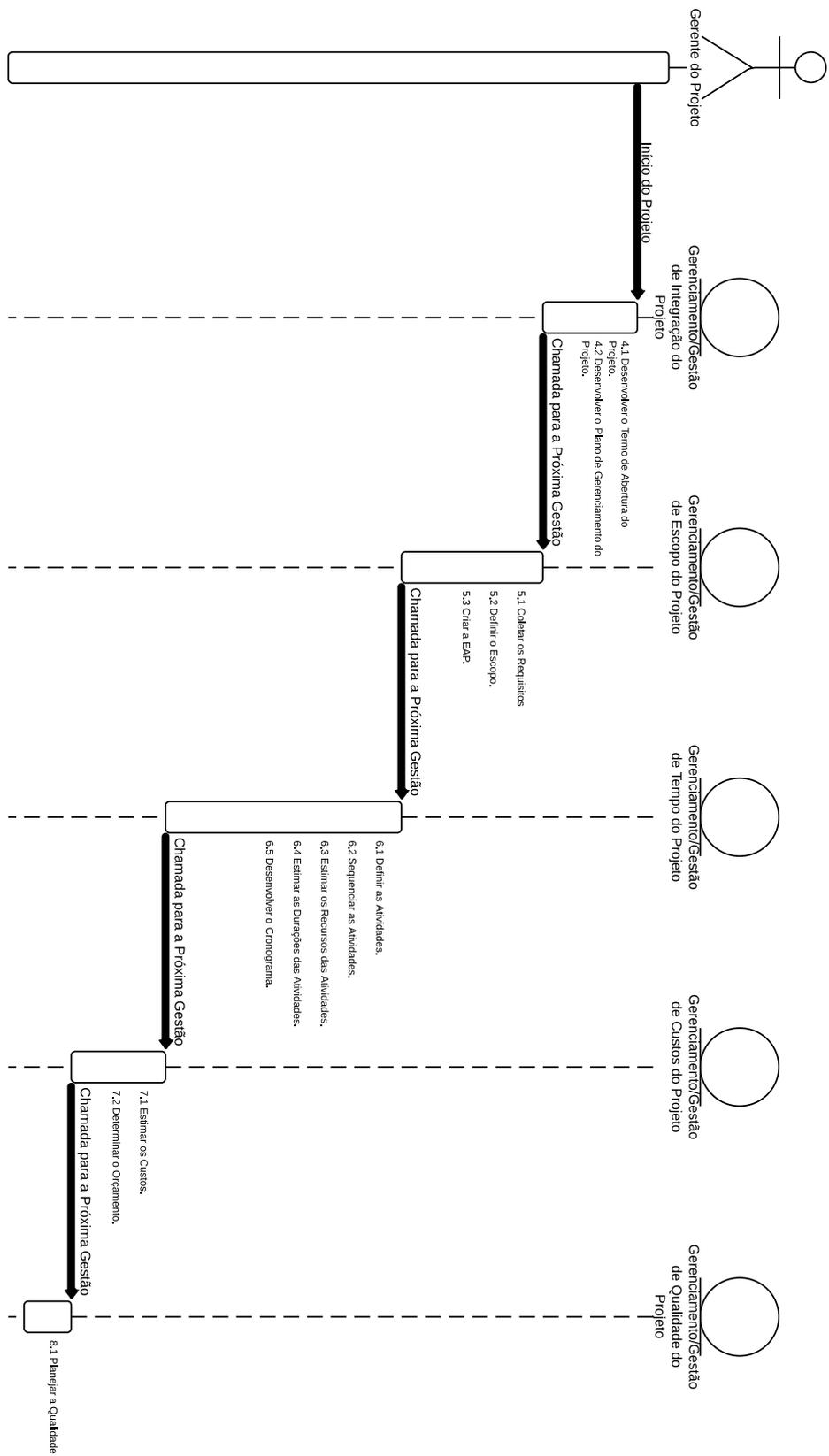


Figura 9 – Modelagem do estado ideal dos processos em diagrama UML sequencial. Fonte: Autor

As comparações entre o modelo atual (4.1.2) e o modelo ideal, previamente citado, serão realizadas na seção de Proposição e Aplicação de Melhorias (4.1.4).

4.1.3.1: Integração

Nesta seção são especificados os processos de Integração contidos nos grupos de Inicialização e Planejamento do modelo ideal (Figura 9).

Desenvolver o termo de abertura do projeto é o processo de desenvolvimento de um documento que formalmente autoriza um projeto ou uma fase e a documentação dos requisitos iniciais que satisfaçam as necessidades e expectativas das partes interessadas. Estabelece uma parceria entre a organização executora e a organização solicitante (ou cliente, no caso de projetos externos). O termo de abertura do projeto formalmente o inicia. Um gerente de projetos é identificado, selecionado e designado o mais cedo possível, preferivelmente enquanto o termo de abertura está sendo desenvolvido e sempre antes do início do planejamento. É recomendado que o gerente de projetos participe do desenvolvimento do termo de abertura, uma vez que este supre o gerente com a autoridade para usar recursos nas atividades do projeto.

Desenvolver o plano de gerenciamento do projeto é o processo de documentação das ações necessárias para definir, preparar, integrar e coordenar todos os planos auxiliares. O plano de gerenciamento do projeto define como o mesmo é executado, monitorado, controlado e encerrado. O conteúdo do plano de gerenciamento do projeto variará dependendo da área de aplicação e complexidade do mesmo. O plano de gerenciamento é desenvolvido através de uma série de processos integrados até o encerramento do projeto. Esse processo resulta num plano de gerenciamento do projeto que é progressivamente elaborado através de atualizações, controladas e aprovadas pelo processo Realizar o controle integrado de mudanças. Pode-se observar a complexa rede de entradas e saídas do processo em questão na figura 10.

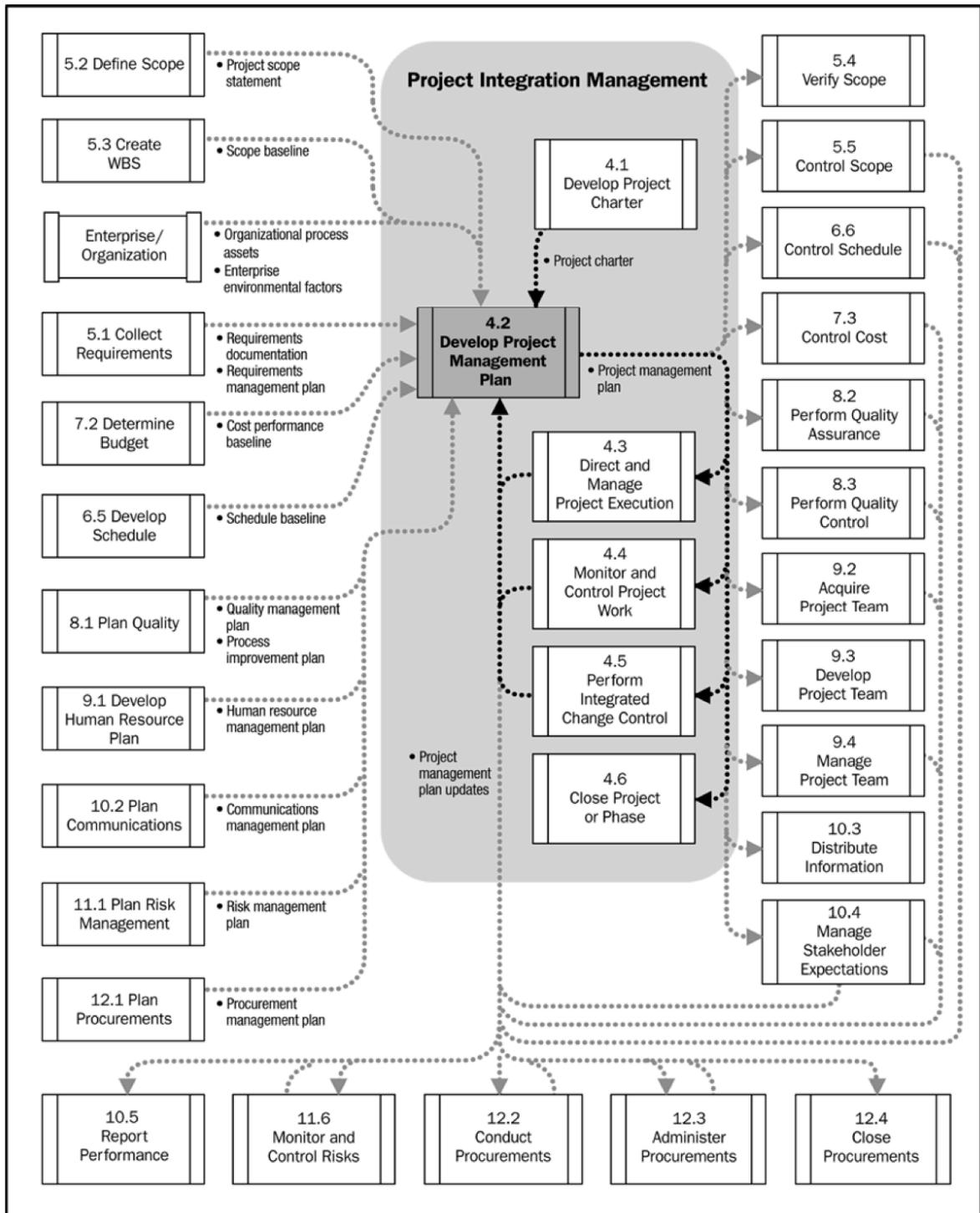


Figura 10 – Diagrama de entradas e saídas do processo Desenvolver o Plano de Gerenciamento do Projeto. Fonte: [1]

4.1.3.2: Escopo

Nesta seção são especificados os processos de Escopo contidos no grupo de Planejamento do modelo ideal (Figura 9).

Coletar os Requisitos é o processo de definir e documentar as funções e funcionalidades do projeto e do produto necessárias para atender as necessidades e expectativas das partes interessadas. O sucesso do projeto é diretamente influenciado pela atenção na captura e gerenciamento dos requisitos do projeto e do produto. Os requisitos incluem as necessidades quantificadas e documentadas, e as expectativas do patrocinador, cliente e outras partes interessadas. Estes requisitos precisam ser obtidos, analisados e registrados com detalhes suficientes para serem medidos uma vez que a execução do projeto se inicie. Coletar os requisitos é definir e gerenciar as expectativas do cliente. Estes requisitos se transformam na fundação da EAP.

Definir o escopo é processo de desenvolvimento de uma descrição detalhada do projeto e do produto. A preparação detalhada da declaração do escopo é crítica para o sucesso e baseia-se nas entregas principais, premissas e restrições que são documentadas durante a iniciação do projeto. Durante o planejamento, o escopo é definido e descrito com maior especificidade conforme as informações a respeito do projeto são conhecidas. Os riscos existentes, premissas e restrições são analisados para verificar sua integridade; riscos adicionais, premissas e restrições são adicionados conforme necessário.

Criar a EAP é o processo de subdivisão das entregas e do trabalho do projeto em componentes menores e de gerenciamento mais fácil. A estrutura analítica do projeto (EAP) é uma decomposição hierárquica orientada às entregas do trabalho a ser executado pela equipe para atingir os objetivos do projeto e criar as entregas requisitadas, sendo que cada nível descendente da EAP representa uma definição gradualmente mais detalhada da definição do trabalho do projeto. A EAP organiza e define o escopo total e representa o trabalho especificado na atual declaração do escopo do projeto aprovada.

4.1.3.3: Tempo

Nesta seção são explicitados os processos de Tempo contidos no grupo de Planejamento do modelo ideal (Figura 9).

Definir as atividades é o processo de identificação das ações específicas a serem realizadas para produzir as entregas do projeto. O processo Criar a EAP identifica as entregas no nível mais baixo da estrutura analítica do projeto (EAP), o pacote de trabalho. Esses pacotes são tipicamente decompostos em componentes menores chamados atividades que representam o trabalho necessário para completar o pacote de trabalho. As atividades proporcionam uma base para a estimativa, desenvolvimento do cronograma, execução e monitoramento e controle do trabalho do projeto. Implícitos neste processo estão a definição e o planejamento das atividades de desenvolvimento do cronograma de tal modo que os objetivos do projeto sejam alcançados.

Sequenciar as atividades é processo de identificação e documentação dos relacionamentos entre as atividades do projeto. Essas são sequenciadas usando relações lógicas. Cada atividade e marco, com exceção do primeiro e do último, são conectados a pelo menos um predecessor e um

sucessor. O uso de tempo de antecipação ou de espera pode ser necessário entre as atividades para dar suporte a um cronograma de projeto realista e executável. O sequenciamento pode ser executado através do uso de software de gerenciamento de projetos ou do uso de técnicas manuais ou automatizadas.

Estimar os recursos da atividade é o processo de estimativa dos tipos e quantidades de material, pessoas, equipamentos ou suprimentos que serão necessários para realizar cada atividade. O processo Estimar os recursos da atividade é estreitamente coordenado junto com o processo Estimar os custos.

Estimar as durações da atividade é o processo de estimativa do número de períodos de trabalho que serão necessários para terminar as atividades específicas com os recursos estimados. A estimativa das durações das atividades utiliza informações sobre as atividades do escopo do projeto, tipos de recursos necessários, quantidades estimadas de recursos e calendários de recursos. As entradas para as estimativas de duração da atividade se originam da pessoa ou grupo na equipe do projeto que está mais familiarizado com a natureza do trabalho na atividade específica. A estimativa da duração é elaborada progressivamente e o processo considera a qualidade e a disponibilidade dos dados de entrada. Por exemplo, conforme o trabalho de engenharia e planejamento do projeto se desenvolve, dados mais detalhados e precisos se tornam disponíveis e a precisão das estimativas de duração melhora. Portanto, a estimativa da duração pode ser assumida como sendo progressivamente mais precisa e de melhor qualidade.

Desenvolver o cronograma é o processo de análise de sequências das atividades, suas durações, recursos necessários e restrições do cronograma visando criar o cronograma do projeto. A entrada das atividades, durações e recursos na ferramenta de elaboração de cronograma gera um cronograma com datas planejadas para completar as atividades do projeto. O desenvolvimento de um cronograma de projeto aceitável é frequentemente um processo iterativo. Determina as datas planejadas de início e de término para as atividades e marcos do projeto. Pode requerer a análise e revisão das estimativas de duração e de recursos para criar um cronograma aprovado do projeto que pode servir como linha de base para acompanhar o seu progresso. A revisão e a manutenção de um cronograma realista continua sendo executada durante todo o projeto à medida que o trabalho progride, o plano de gerenciamento do projeto muda e a natureza dos eventos de riscos evolui.

4.1.3.4: Custos

Nesta seção são explicitados os processos de Custos contidos no grupo de Planejamento do modelo ideal (Figura 9).

Estimar os custos é o processo de desenvolvimento de uma estimativa dos recursos monetários necessários para executar as atividades do projeto. As estimativas de custo são um prognóstico baseado na informação conhecida num determinado momento. Incluem a identificação e a consideração das alternativas de custo para iniciar e terminar o projeto. Compensações de custos e

riscos devem ser consideradas, como fazer versus comprar, comprar versus alugar e o compartilhamento de recursos para alcançar custos otimizados para o projeto.

Determinar o orçamento é o processo de agregação dos custos estimados de atividades individuais ou pacotes de trabalho para estabelecer uma linha de base dos custos autorizada. Essa linha de base inclui todos os orçamentos autorizados, mas exclui as reservas de gerenciamento.

4.1.3.5: Qualidade

Nesta seção são modelados os processos de Qualidade contidos no grupo de Planejamento do modelo ideal (Figura 9).

Planejar a qualidade é o processo de identificação dos requisitos e/ou padrões de qualidade do projeto e do produto, além da documentação de como o projeto demonstrará a conformidade. O planejamento da qualidade deve ser realizado em paralelo com os outros processos de planejamento do projeto. Por exemplo, modificações propostas no produto para atender aos padrões de qualidade identificados podem exigir custos ou ajustes nos cronogramas e uma análise de riscos detalhada dos seus impactos nos planos.

4.1.4: Proposição e Aplicação de Melhorias

Para propor e aplicar as melhorias, relacionadas a gestão de projetos, na empresa Cheesecake Labs, escolheu-se um projeto realizado com a Cogentio – gerenciado pela TriplePoint – para servir de caso de estudo. No projeto escolhido a intenção dos clientes é reformular uma existente página web com um novo design e algumas novas funcionalidades. A página web em questão é utilizada pelos funcionários da empresa TriplePoint para acesso e edição do conteúdo de seus banco de dados.

As melhorias são geradas a partir da comparação entre o modelo atual e ideal, em conjunto com as necessidades dos gestores da Cheesecake Labs e da Cogentio.

4.1.4.1: Proposta do Cliente e *Business Case*

As mudanças na página de *admin* da Cogentio tem como função modificar a forma de disponibilização e edição das informações de seus banco de dados, *renderizando*-as em tabelas com ferramentas para editar seu conteúdo de maneira prática. Serão utilizadas ferramentas de *renderização* de *templates* no *front-end* para possibilitar melhor escalabilidade de processamento. Dessa maneira, os clusters de servidores não precisam gastar tempo realizando a função de *renderizar* tabelas, pois os *browsers* dos usuários fazem essa tarefa.

As mudanças do site não se limitam à página de tabelas, pois também requerem mudanças no *layout* da página de *login* e também na página inicial (*home*). O *fluxo de telas* da versão antiga da página de *admin* da Cogentio é apresentado nas figuras 11 e 12.

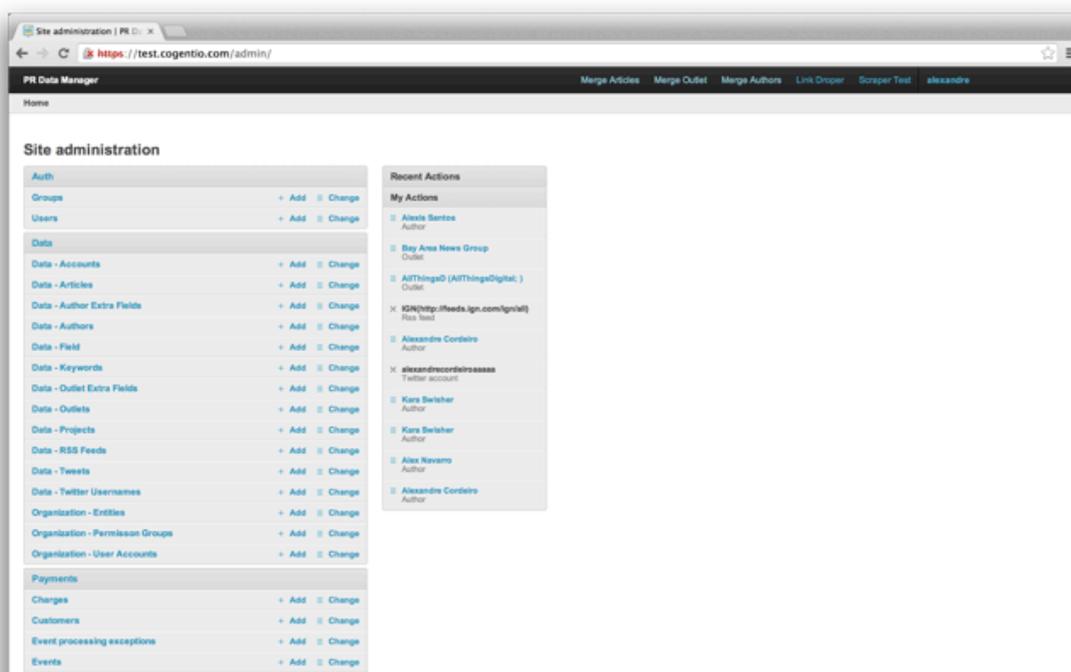


Figura 11 – Versão antiga da página inicial (*home*) da Cogentio. Fonte: Autor

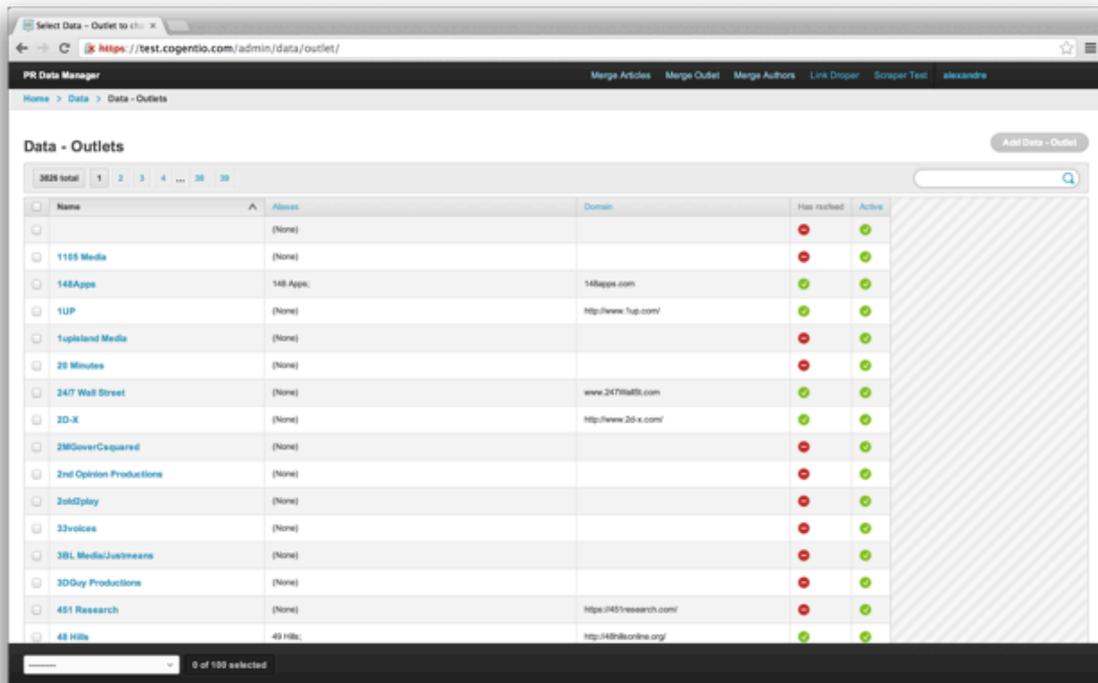


Figura 12 – Versão antiga da página de visualização de tabelas da Cogentio.

Fonte: Autor

Os clientes, nesse caso, criaram *Mock-ups* para representar o que deve ser feito, em termos gráficos. Os *Mock-ups* referentes às páginas *home* e de visualização de tabelas estão apresentados nas figuras 13 e 14:

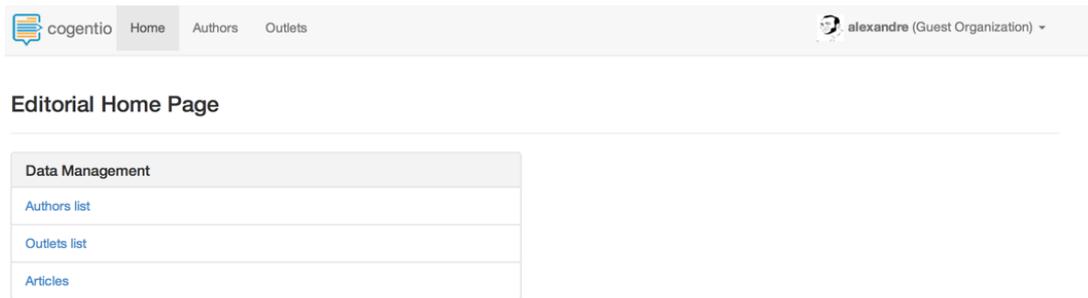


Figura 13 – Mock-up da nova tela de home. Fonte: Autor

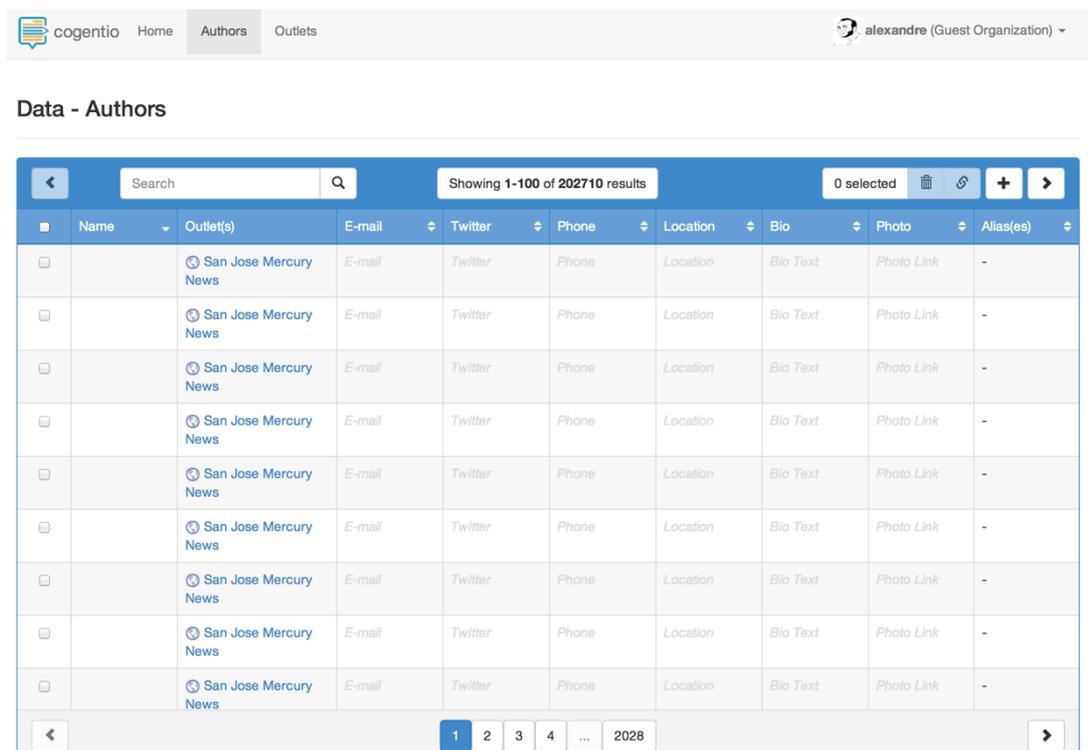


Figura 14 – Mock-up da nova tela de visualização de tabelas. Fonte: Autor

As *features*, ou requisitos de produto, definidas nos documentos com a Cogentio são:

- *Template* de web design responsivo e adaptável às telas de *tablets* e celulares.

- Perfil de usuário com foto e nome da organização.
- Opção de reorganizar tabelas baseando-se em atributos.
- Opção de fundir (integrar os conteúdos de) artigos das tabelas, com o intuito de remover informações duplicadas.
- Adicionar e remover campos do banco de dados nas tabelas "renderizadas" da classe *Outlet*.

Os requisitos de projetos para a implementação das *features* são:

- Integrar os elementos do existente website com o conteúdo dos *templates* de *web design* para as seguintes páginas: *login*, *home* e tabela de visualização.
- Alterar o modelo do banco de dados para a classe Usuário adquirir os novos atributos: *url* da imagem de capa e *string* da organização.
- Instanciar e configurar servidor de S3 da Amazon para hospedar as imagens.
- Criar interface de upload de imagens.
- Integrar servidor EC2, em Python, com os serviços da S3.
- Criar métodos JavaScript para reorganizar conteúdo de tabelas em função de um atributo.
- Criar métodos Javascript para integrar as chamadas de fusão de artigos com a interface das tabelas.
- Adicionar os campos na *renderização* das tabelas de *outlets*: Twitter, E-mail, Localização, Bio., Logo e Autores.
- Remover o campo Ativo na *renderização* das tabelas de *outlets*.
- Mover as tarefas de *renderização* de *templates* para o *front-end*, por questões de escalabilidade, utilizando a biblioteca React.js.

4.1.4.2: Melhorias na Estrutura Analítica do Projeto (EAP)

A maior dificuldade detalhada pelo corpo gerencial da Cheesecake Labs, se tratando de gestão de projetos, foi a sua inexperiência em lidar com a estrutura analítica do projeto, ou seja: a relação entre as *features* e requisitos de projetos não se fazia clara, causando uma desorganização na subdivisão e sequenciamento de atividades.

A subdivisão e sequenciamento das tarefas era feita de maneira informal e, em muitos casos, sem o consenso de todas as partes envolvidas. Em conjunto com a gestão da Cheesecake Labs e se

baseando no Guia PMBoK, foi decidido que adotar a utilização de uma ferramenta de WBS (*Work Breakdown Structure*) seria essencial para solucionar os problemas previamente descritos.

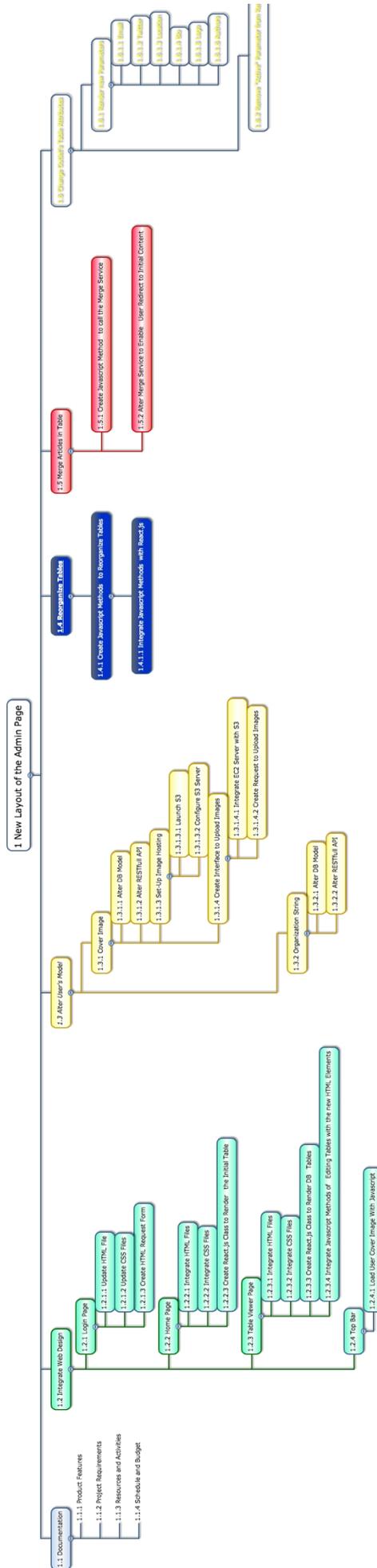
O WBS soluciona o problema descrito previamente pois organiza a relação entre *features* e requisitos, explicitando também suas dependências. Dessa maneira, torna-se possível uma objetiva comunicação com os clientes se tratando da subdivisão de tarefas, de seu sequenciamento e, posteriormente, do cronograma.

A metodologia de escolha da ferramenta de modelagem foi buscar na *web* quais as referências que possuíam os recursos necessários e fossem livre de custos. Os recursos que foram buscados nas ferramentas são:

- Capacidade de trabalhar em um arquivo de WBS em equipe (de maneira distribuída).
- Interface intuitiva e completa.
- Capacidade de exportar o documento como imagem.

A ferramenta que melhor se enquadrou no perfil desejada foi a WBSTool [17], que é de uso gratuito, possui interface de edição na *web* – possibilitando o trabalho distribuído – e tem todas as notações e elementos lógicos para criar o WBS.

Utilizando o WBSTool, foi criada – em parceria com o corpo gestor da Cheesecake Labs – a estrutura analítica de projeto para o caso de estudo de remodelagem do site *admin* da Cogentio. O resultado da modelagem da estrutura analítica do projeto é apresentado pela figura 15.



www

Figura 15 – Modelagem da estrutura analítica de projeto. Fonte: Autor

Pode-se observar que o modelo WBS representa de maneira intuitiva a relação entre as *features* de produto e os requisitos de projeto, explicitando suas dependências e subdivisões. Dessa maneira, torna-se mais clara e objetiva a compreensão da estrutura do trabalho a ser realizado.

4.1.4.3: Melhorias no Desenvolvimento do Cronograma

Como consequência do melhor planejamento da estrutura analítica do projeto, descrito na seção anterior, tornou-se mais clara a necessidade de melhorias nos processos de desenvolvimento de cronograma. Na situação inicial da empresa os cronogramas eram desenvolvidos de maneira informal e mal detalhada.

O corpo gerencial da Cheesecake confirmou que o cronograma era um ponto de tensão com os clientes na maioria dos projetos. As datas de entrega eram raramente cumpridas e o escopo das tarefas possuíam constantes mudanças.

Para solucionar os problemas previamente citados, conclui-se ser essencial a utilização de uma ferramenta e norma de modelagem para melhor organizar graficamente o cronograma dos projetos. O Guia PMBoK, na área de conhecimento de gestão de tempo, cita a utilização de diagramas de Gantt para organizar o cronograma do projeto. Dessa maneira, iniciou-se a busca por uma ferramenta de modelagem de diagramas de Gantt.

A metodologia de escolha da ferramenta de modelagem foi buscar na *web* quais as referências que possuíam os recursos necessários e baixo custo. Os recursos que foram buscados nas ferramentas são:

- Capacidade de trabalhar em um arquivo de diagrama Gantt em equipe (de maneira distribuída).
- Interface intuitiva e completa.
- Capacidade de atualizar o progresso do diagrama ao longo do desenvolvimento do projeto de maneira simples e prática.
- Capacidade de exportar o documento como imagem.

A ferramenta que melhor satisfaz as necessidades da Cheesecake Labs foi o TeamGantt [18], que possui interface de edição na *web* – possibilitando o trabalho distribuído, interface intuitiva, baixos custos mensais e funcionalidades que facilitam a atualização do progresso das tarefas. A capacidade de atualizar o progresso das tarefas é um fator decisivo do TeamGantt pois facilita, além do planejamento, o controle futuro do cronograma que é realizado ao longo do desenvolvimento do projeto.

Para a criação do diagrama de Gantt utilizou-se a estrutura analítica do projeto (citada na seção anterior) como entrada, em conjunto com as estimativas de duração de cada atividade (foram

previamente realizadas pela equipe da empresa). O diagrama modelado é representado pela figura 16.

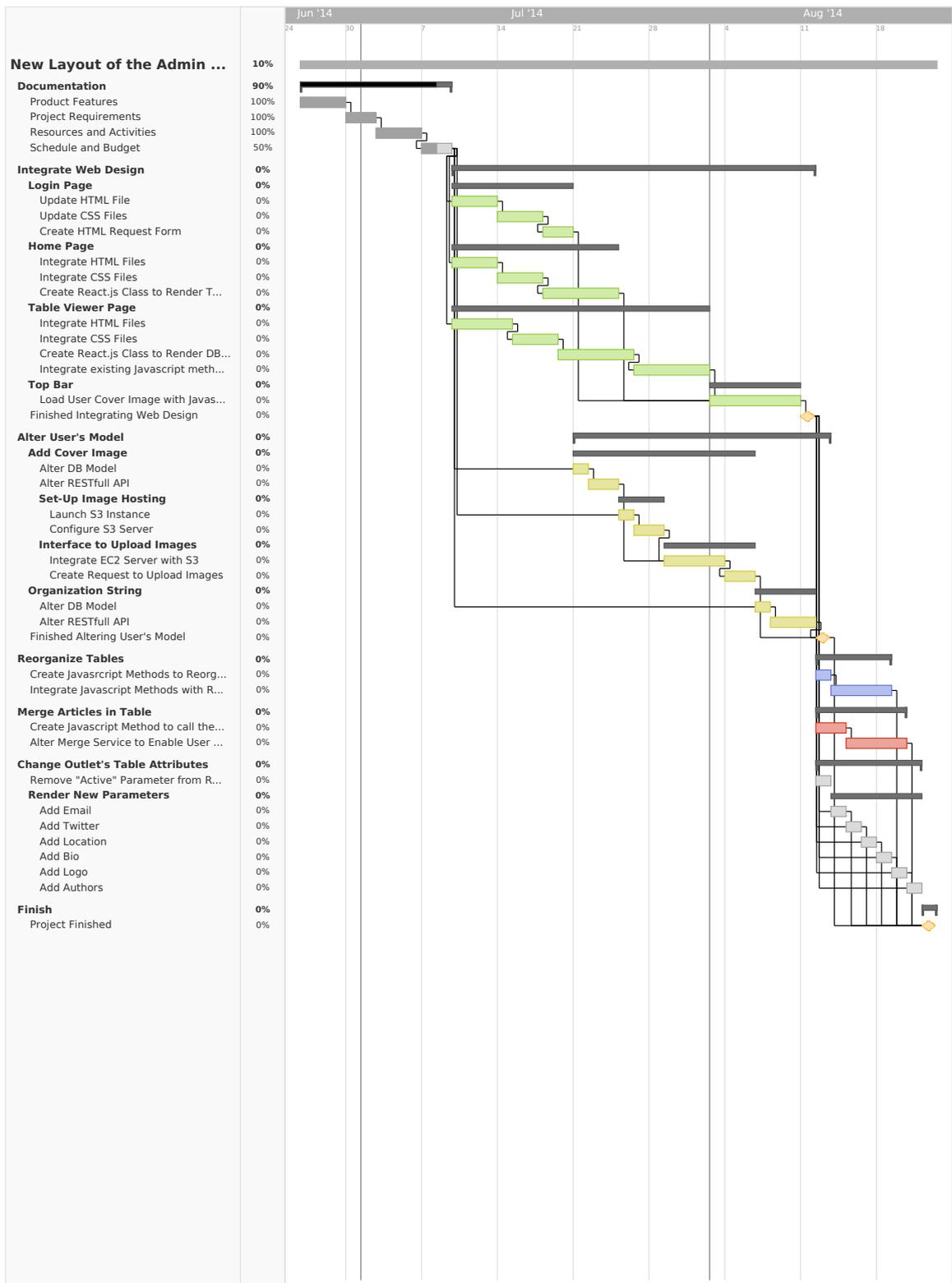


Figura 16 – Diagrama de Gantt das tarefas a serem realizadas no projeto com a Cogentio. Fonte: Autor

4.1.4.4: Melhorias na Determinação do Orçamento

O processo de determinar orçamento também foi definido como crítico pelos gerentes da Cheesecake Labs, pois em projetos de longo prazo os clientes perdem o controle de onde seus recursos estão sendo realmente investidos. Para tornar os documentos de orçamento mais claros e objetivos foi escolhido o método de custeio baseado em atividades, ou custeio ABC, proposto pelo Guia PMBoK em sua área de conhecimento de gestão de custos.

O custeio ABC é um método de custeio baseado nas atividades que a empresa realiza no processo de desenvolvimento de seu produto. O objetivo do custeio ABC é fornecer um método para o tratamento dos custos indiretos, através da análise das atividades e de seus geradores de custo.

No caso de estudo aplicou-se o custeio ABC no orçamento do projeto, para que a Cogentio pudesse ter mais controle de como seus recursos estavam sendo investidos nas diversas funcionalidades desenvolvidas. Para melhor descrever os custos indiretos do orçamento, utilizou-se com entrada (*input*) as informações definidas pela estrutura analítica do projeto, em conjunto com a duração das atividades e o custo dos recursos envolvidos.

Correlacionando a duração do desenvolvimento dos requisitos – que compõe as *features* do produto – com o preço dos recursos envolvidos, foi possível calcular o custo de desenvolvimento aproximado de cada uma das *features*. Os recursos envolvidos no projeto são em maioria desenvolvedores e gestores que possuem seu valor cobrado por hora.

Pode-se observar o antigo e o novo formato de orçamento nas tabelas 9 e 10.

Tabela 9 – Antigo formato de orçamento. Fonte: Autor

Days	Worked Hours	Cost
86	688	\$27,520.00

Tabela 10 – Novo formato de orçamento, utilizando custeio ABC. Fonte: Autor

Activities	Days	Worked Hours	Cost
Documentation	10	80	\$3,200.00
Integrate Web Design			
Login	8	64	\$2,560.00
Home	11	88	\$3,520.00
Table	17	136	\$5,440.00
Top Bar	6	48	\$1,920.00
Alter User's Model			
Cover Image	12	96	\$3,840.00
Organization String	4	32	\$1,280.00
Reorganize Tables	5	40	\$1,600.00
Merge Articles	6	48	\$1,920.00
Update Outlet's Table	7	56	\$2,240.00
Total	86	688	\$27,520.00

4.2: Gestão de Desenvolvimento de *Software*

Nesta seção serão realizadas as etapas descritas no começo do capítulo para os processos de gestão de desenvolvimento de *software*. Devido à especificação clara e objetiva dos gerentes da Camiolog – tratando da questão de testes – essa seção será mais focada em buscar, no CMMI-DEV, embasamento para a realização dos mesmos.

Diferentemente da seção anterior (4.1), a modelagem do sistema ideal de Gestão de Desenvolvimento de Software será apenas utilizada para localizar e especificar, no CMMI-DEV, o posicionamento dos processos de testes dentro do contexto geral do projeto. A característica previamente citada faz com que esta seção seja mais enxuta, com relação a anterior, devido ao foco dos clientes em processos específicos de testes.

4.2.1: Escolha da Metodologia

Para a gestão de desenvolvimento de software foi escolhido o CMMI-DEV por já estar consagrado no mercado norte americano, mesmo sendo uma alternativa mais cara que o MPS.BR. Essa escolha se deve ao fato de como os clientes da Cheesecake Labs estão localizados nos EUA, faz mais sentido buscar uma solução que será melhor compreendida e recebida por eles.

4.2.2: Modelagem do Estado Atual

O grande tema em questão levantado pelos gerentes da Camiolog, tratando-se de processos críticos de desenvolvimento de software, foi a necessidade de testes para o aplicativo de câmera.

Como em primeira instância nenhum teste existia, o modelo do estado atual do sistema não se faz relevante.

4.2.3: Modelagem do Estado Ideal

Na norma CMMI-DEV, o processo que melhor representa a necessidade crítica dos gerentes da Camilog é o "Solução Técnica", de maturidade nível 3. O processo de Solução Técnica engloba a realização das seguintes tarefas:

1. Selecionar Soluções para os Componentes do Produto
 - 1.1. Desenvolver Diferentes Soluções e Critérios de Seleção.
 - 1.2. Selecionar as Soluções para os Componentes do Produto.
2. Desenvolver o Projeto
 - 2.1 Projetar o Produto ou Componente de Produto.
 - 2.2 Estabelecer um Pacote de Dados Técnicos.
 - 2.3 Projetar as Interfaces Utilizando Critérios.
 - 2.4 Realizar análises de "Fazer", "Comprar" ou "Reutilizar".
3. Implementar o Projeto do Produto
 - 3.1 Implementar o Projeto.
 - 3.2 Desenvolver Documentação de Suporte ao Produto.

Para simplificar o problema, foi escolhido para estudo o processo que melhor se encaixa na necessidades dos gerentes da Camilog: "Implementar o Projeto", inclui a alocação, refinamento e verificação de cada componente do produto. Também envolve a coordenação entre os vários esforços de desenvolvimento de componente do produto com o intuito de, no caso da Camilog, codificar o software.

A CMMI-DEV descreve um conjunto de práticas que devem pautar o processo de "Implementar o Projeto":

1. Utilizar métodos efetivos para implementar os componentes do produto:
 - Programação Estruturada;
 - Programação Orientada a Objeto;
 - Programação Orientada a Aspecto;
 - Geração Automática de Código;
 - Reutilização de Código;

- Utilização de Padrões de Design;
2. Utilizar Padrões e Critérios:
 - Padrões de Linguagem;
 - Requerimentos em Forma de Diagramas;
 - Padrões de Peças;
 - Estrutura e Hierarquia de Componentes de Software;
 - Critérios:
 - Modularidade;
 - Clareza;
 - Simplicidade;
 - Confiabilidade;
 - Segurança;
 - Facilidade de Suporte;
 3. Conduzir revisão por pares nos componentes de produtos selecionados;
 4. Realizar testes unitários dos componentes de produtos apropriados:
 - Cobertura de Testes por Afirmação;
 - Cobertura de Testes por Galhos;
 - Cobertura de Testes por Predicados;
 - Cobertura de Testes por Caminho;
 - Testes de Fronteiras de Valores;
 - Testes de Valores Especiais;
 5. Revisar os Componentes do Produto;

4.2.4: Proposição e Aplicação de Melhorias

As melhorias descritas neste são geradas a partir da comparação entre o modelo atual e ideal, em conjunto com as necessidades dos gestores da Cheesecake Labs e da Camiolog.

Para propor e aplicar as melhorias foi escolhida a seção de testes unitários, dentro do processo de "Solução Técnica" – descrito pela norma CMMI-DEV. Esta seção tem como intuito criar testes unitários para cenários críticos no funcionamento do aplicativo de câmera da Camiolog. Os

testes serão criados utilizando como ambiente de desenvolvimento o Eclipse – por ser o software adotado pela Cheesecake – em conjunto com o kit de desenvolvimento do Android. Como a linguagem dos testes de Android será Java, a biblioteca de testes utilizada será padrão, o JUnit.

Para a realização dos testes unitários de maneira escalável é essencial criar uma estrutura lógica que remova a dependência do aplicativo de repostas da rede, devido ao seu longo tempo de espera. Dessa maneira, a primeira tarefa para implementar os testes unitários é criar uma estrutura que simula a camada de rede (*mock*). Pode-se observar a estrutura de testes, com a conexão de rede *mockeada*, na figura 17.

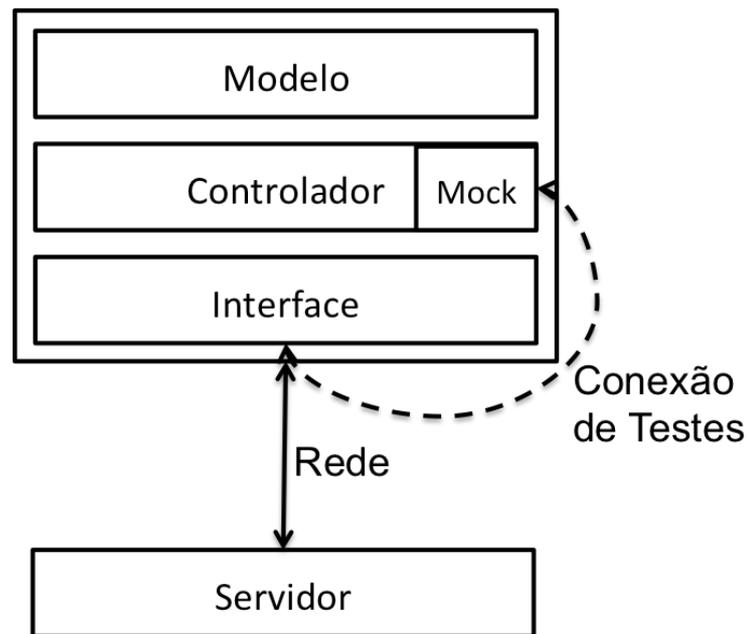


Figura 17 – Estrutura da software da Camiolog, salientando a conexão de rede mockeada (falsa) de testes. Fonte: Autor

A metodologia de busca por uma biblioteca responsável pela criação da estrutura de Mocks da camada de rede foi procurar no website Github [19] quais eram os repositórios de versionamento que possuíam as seguintes características:

- Usuários ativos que continuam mantendo o *software* atualizado;
- Grande base de usuários (para garantir estabilidade);
- Utilização intuitiva;
- Sintaxe organizada e simples.

A biblioteca que melhor satisfaz as necessidades citadas previamente foi a WireMock [20], que permite a criação de classes de rede falsas em instantes e possui 122 versões mantidas por diferentes desenvolvedores. Pode-se observar um exemplo de utilização da WireMock na figura 18.

```
@Test
public void exactUrlOnly() {
    stubFor(get(urlEqualTo("/camiolog/"))
        .willReturn(aResponse()
            .withHeader("Content-Type", "text/plain")
            .withBody(""));
    assertThat(testClient.get("/camiolog").statusCode(), is(200));
    assertThat(testClient.get("/camiolog/unwanted/url").statusCode(), is(404));
}
```

Figura 18 – Exemplo de utilização da biblioteca WireMock para criar uma estrutura falsa de rede (em Java). Fonte: Autor

Recentemente a Camiolog desenvolveu um sistema de autenticação que envolve *tokens* de upload que gerenciam a validação, *back-end*, das requisições vindas do aplicativo da câmera. Essa nova funcionalidade criou um cenário de risco no qual se, por algum motivo, o *token* de *upload* da câmera se tornar inválido, suas imagens não serão mais enviadas para o servidor – criando um situação de crítica para os usuários. Para garantir que o problema previamente descrito não aconteça, é essencial que o aplicativo da câmera tente adquirir outro *token* de *upload* caso o seu se torne inválido.

Devido ao cenário crítico previamente citado, os gerentes da Camiolog decidiram que os primeiros testes unitários desenvolvidos deveriam garantir que a nova funcionalidade – de expiração dos *tokens* de *upload* – funcionem de maneira robusta. Dessa forma, foi implementado um teste que garante que o aplicativo tenta gerar outro *token* caso receba uma resposta HTTP de número 401, indicando falha na autenticação. A implementação do teste pode ser vista na figura 19.

```

@Test
public void testUpload401() {
    // Creates a stub that returns 401 when camera tries to upload image.
    stubFor(get(urlEqualTo("/camiolog/api/upload"))
        .willReturn(aResponse().withStatus(401));

    // Sets the stub as the default HTTP client.
    CamiologHttpClient.setDefaultClient(testClient);

    // Makes sure HTTP client is idle.
    assertEquals(CamiologHttpClient.getStatus(), CamiologHttpClient.IDLE);

    // Makes sure user's token is not null.
    assertNotNull(User.uploadToken);

    // Forces camera to upload last frame.
    CamiologHttpClient.forceUploadWithLastFrame();

    // Makes sure user's token was cleared.
    assertNull(User.uploadToken);

    // Makes sure HTTP client is trying to authenticate.
    assertEquals(CamiologHttpClient.getStatus(), CamiologHttpClient.AUTHENTICATING);
}

```

Figura 19 – Implementação do teste de rede utilizando a biblioteca WireMock.

Fonte: Autor

Capítulo 5: Análise Crítica

No presente capítulo é realizada uma análise crítica das metodologias e resultados explicitados no capítulo anterior. A análise tem como objetivo entender quais são os pontos positivos e negativos dos desenvolvimentos realizados nos dois casos de estudo. Para melhor compreender as mudanças no contexto gerencial da Cheesecake Labs, são analisadas as metodologias utilizadas para: escolher normas e guias; escolher ferramentas; realizar a modelagem do estado atual e ideal dos processos e propor (e implementar) melhorias.

Neste capítulo também são analisados os resultados reais das melhorias implementadas nos processos críticos dos casos de estudo. Dessa maneira, é explicitada a relação entre os pontos positivos e negativos das metodologias utilizadas neste trabalho, e suas consequências nos resultados reais dos processos (em ambos os casos de estudo).

5.1: Gestão de Projetos.

A escolha das normas e metodologias de gestão de projeto fez com que o escopo de estudo do trabalho se tornasse muito amplo, devido a imensa quantidade de conteúdo proveniente do Guia PMBoK e ISO 21500. Essa característica da metodologia fez com que muito tempo fosse investido estudando conteúdos que – em alguns casos – nem foram utilizadas para o desenvolvimento do trabalho. Esse fato é proveniente da inexperiência dos gestores da Cheesecake Labs, pois como não conheciam as principais normas vigentes de Gestão de Projetos, se fez essencial um estudo mais amplo e introdutório.

Os esforços futuros, da Cheesecake Labs, com o objetivo de melhorar os processos de gestão de projetos não terão os problemas citados no parágrafo anterior, pois agora o corpo gerencial possui conhecimento das principais normas e é capaz de focar seus objetivos em questões mais específicas e bem documentadas.

A escolha de ferramentas foi o processo menos documentado devido ao seu caráter empírico. O Guia PMBoK não explicita quais ferramentas devem ser utilizadas para a real implementação dos modelos e diagramas, descritos no mesmo. Dessa maneira, a escolha de ferramentas deve ser experimental, fazendo com que seu aprimoramento tenha que ser pautado por experiências práticas.

5.1.1: Análise das Ferramentas

A análise das ferramentas pode ser dividida nos exemplos utilizados no estudo de caso:

- **LucidChart**

A ferramenta LucidChart tem como pontos positivos: flexibilidade, por possuir notações de diversas normas; facilidade de uso, devido a estrutura de Javascript coesa que define a interface e capacidade de trabalho distribuído, por ser implementada como um serviço *web*.

Os pontos negativos da ferramenta LucidChart são: incapacidade de versionamento, pois cada arquivo possui apenas o último estado salvo no servidor; incapacidade de especificar colaboradores, pois como cada arquivo é acessado por apenas uma conta, os logs não possuem identificação de quem realizou as mudanças e, finalmente, incapacidade de modelar arquivos grandes, devido à limitação de memória imposta pelo navegador (*browser*).

- **WBSTool**

A ferramenta WBSTool tem como pontos positivos: utilização gratuita; capacidade de trabalhar em arquivos de maneira distribuída e biblioteca de elementos gráficos completa. Os pontos negativos da WBSTool são: interface não intuitiva para edição de diversos elementos ao mesmo tempo; incapacidade de criar arquivos muito grandes e, finalmente, falta de conteúdo online para guiar seu desenvolvimento.

- **TeamGantt**

A ferramenta TeamGantt tem como pontos positivos: a facilidade de manter o progresso das atividades atualizado; interface simples e intuitiva e capacidade de trabalho distribuído. Seus pontos negativos são: preço dependente do número de contas com acesso aos arquivos, incapacidade de gerenciar versionamento e limitação do tamanho projeto (devido à memória do navegador).

A metodologia de modelagem do estado atual do sistema também possui características empíricas devido ao aspecto experimental de criação dos processos, na Cheesecake Labs. A norma PMBoK guiou os esforços, de modelagem do estado atual, no sentido de definir qual o escopo das tarefas e quais informações deveriam ser coletadas e organizadas. A notação BPMN também foi útil para tornar mais padronizada e organizada a modelagem dos processos gerenciais.

A metodologia de modelagem do estado ideal do sistema necessitou grandes esforços de compreensão do PMBoK, pois o estado ideal descrito pelo Guia é complexo e possui muitos conteúdos que, em alguns casos, nem se aplicam ao contexto de *software* de maneira clara.

Os esforços de compreensão do Guia PMBoK trazem muitos resultados positivos pois seu conteúdo embasa as necessidades críticas dos projetos: dando mais segurança ao gestor, por organizar e documentar questões, antigamente definidas como, informais; clarificando quais devem ser os caminhos a serem trilhados pela empresa (no contexto de melhoria de processos de Gestão de Projetos), por saber quais os pontos a serem melhorados futuramente e, finalmente, por citar metodologias e normas muito úteis nos processos de gestão.

A metodologia de geração e aplicação de melhorias foi pautada pelas necessidades dos gerentes da Cogentio, ou seja, os esforços buscaram, no Guia PMBoK, base para lidar com questões

críticas salientadas pelos clientes. Dessa forma, a escolha dos pontos que necessitavam melhorias foram definidos por agentes externos ao trabalho, e a solução desses problemas foi buscada no PMBoK. Essa característica faz com que os esforços de geração e aplicação de melhorias conectem realidades distintas: os objetivos práticos da empresa parceira, a capacidade e necessidade da Chesecake Labs e o conteúdo teórico do Guia PMBoK.

Devido ao caráter distribuído das informações (nos terceiros, na Chesecake e no PMBoK) foi essencial realizar as tarefas de maneira iterativa, levando em consideração as opiniões das diferentes partes envolvidas e a base teórica do Guia. O perfil iterativo dos processos fez com que o desenvolvimento se desse de forma gradual, para garantir que todas as partes estavam cientes dos esforços e objetivos de cada etapa. Essa característica do desenvolvimento, em conjunto com as necessidades subjetivas dos clientes, faz com que o tempo de duração da geração e aplicação das mudanças se torne longo e mutável.

5.1.2: Avaliação dos Resultados das Melhorias

A análise dos resultados das melhorias aplicadas ao projeto pode ser dividida nas mudanças realizadas:

- **Melhorias na Estrutura Analítica do Projeto (EAP):**

As melhorias realizadas na Estrutura Analítica de Projeto se tornaram muito úteis para a aplicação das outras mudanças, pois a organização das dependências entre os requisitos de projeto, e suas relações com as *features* do produto, serve de base para sequenciar a realização das atividades, determinar suas durações e melhor organizar a divisão dos esforços com relação às necessidades dos clientes.

Os gerentes da Cogentio consideram as melhorias na EAP o ponto de maior importância nas mudanças realizadas nos processos, pois sua implementação: cria um cenário de comunicação com mais clareza; organiza a definição do produto e, como consequência, o escopo do projeto; facilita a criação de cronogramas e, finalmente, ajuda a dividir e especificar onde os recursos estão sendo investidos.

A parte negativa das melhorias na EAP são referentes ao tempo extra que deve ser despendido para criação do modelo (fator que se torna menos impactante quando os gerentes criarem mais familiaridade com a ferramenta) e à possível imprecisão da linguagem utilizada no mesmo. Para solucionar os problemas de imprecisão da linguagem do modelo WBS, o Guia PMBoK sugere a criação de um dicionário WBS que deve acompanhar o diagrama. O dicionário deve conter um melhor detalhamento dos termos utilizados no diagrama e uma melhor definição de seus possíveis sub tópicos. Dessa maneira, o sistema se torna menos suscetível a erros de interpretação de linguagem e falta de comunicação e clareza.

- **Melhorias no Desenvolvimento do Cronograma:**

As melhorias realizadas no desenvolvimento do cronograma foram úteis para deixar claro, para os gerentes da Cogentio, quais eram os recursos humanos alocados em cada etapa do projeto e qual a previsão de finalização das diversas etapas do projeto. Como a fase de execução do projeto não faz parte do escopo do trabalho, é essencial deixar claro que a função do cronograma vai muito além de prever qual será a data de finalização do projeto, pois também tem como função manter o projeto dentro das expectativas temporais e, eventualmente, realizar mudanças (de realocação de recursos, ou de sequenciamento de atividades), para que os diversos prazos sejam cumpridos.

A definição do cronograma faz com que as *features* e os requisitos devam ser precisamente levantados no início do projeto, pois impactam diretamente em datas e prazos documentados. Essa característica faz com que mais tempo deve ser dispendido, em conjunto com todas as partes envolvidas, nas etapas iniciais do projeto. Dessa maneira, o trabalho se dá de forma organizada e os engenheiros têm uma visão mais clara de qual é a sequência de suas atividades e seus respectivos prazos de término.

- **Melhorias na Determinação do Orçamento:**

As melhorias na determinação do orçamento, utilizando custeio ABC, foram bem vistas pelos gerentes da Cogentio, por representarem uma busca por maior clareza, se tratando de custos. A planilha de custos ABC, desenvolvida neste trabalho, foi mostrada para a banca de investidores da TriplePoint (empresa dona da Cogentio). A banca relatou que as tabelas ajudam no planejamento de longo prazo, pois agregam valor ao correlacionar, de maneira simples, quais os recursos financeiros exigidos para o desenvolvimento das diferentes *features* planejadas. Dessa forma, a banca de investidores da TriplePoint pode repensar a necessidade das *features*, baseando-se no seu custo (outras questões e documentos também estão envolvidos nas decisões para trazer informações sobre outras áreas, como por exemplo: tempo, escopo, integração, entre outros).

O custeio ABC também pode ser aplicado nos custos do projeto, para que a Cheesecake Labs tenha maior controle sobre seus custos indiretos. Dessa maneira, a banca da Cheesecake teria melhor controle sobre como os custos indiretos compõe as despesas totais. Para a realização de um custeio ABC nos custos dos projetos da Cheesecake Labs, pode-se utilizar como entrada (*input*) o custeio ABC que já foi feito para o orçamento, em conjunto com os custos de aluguel, água, luz, internet e as outras despesas operacionais.

5.2: Gestão de Desenvolvimento de *Software*

A escolha das normas e metodologias de gestão de Desenvolvimento de *software* fez com que o escopo do estudo se tornasse amplo, pois as normas CMMI e MPS.br possuem muito conteúdo variado. O estudo das metodologias foi limitado pela necessidade específica dos gerentes da Camilog, ou seja, a clareza da necessidade de testes nos processos simplificou o processo de estudo das normas.

Ainda existem dúvidas se a escolha da CMMI foi a decisão correta, devido ao escopo limitado do estudo realizado neste trabalho. A CMMI possui uma estrutura que dificulta sua utilização em pequenas equipes, isso acontece devido ao alto nível de complexidade gerencial que deve ser alcançado para se atingir os níveis de maturidade. A MPS.br é mais focada no cenário brasileiro e, por possuir etapas de validação menores, pode ser mais simples de ser implementada em um cenário de *startup*.

5.2.1: Análise da Escolha das Ferramentas

A análise da escolha das ferramentas e bibliotecas para criação dos testes pode ser dividida em suas referências:

- **Eclipse**

Escolheu-se o ambiente de desenvolvimento integrado (IDE) Eclipse, pois já estava sendo utilizado por alguns desenvolvedores da Cheesecake Labs. Dessa maneira não seria necessário nenhum novo aprendizado de IDE para a codificação dos testes. A utilização do Eclipse possui como pontos positivos sua capacidade de ser executado em diferentes sistemas operacionais (mantendo o desenvolvimento da equipe íntegro) e também a facilidade de integração com as ferramentas do Android.

Os pontos negativos da utilização do Eclipse foram em maioria relacionados com a falta de velocidade do programa. Por ser executado “em cima” da máquina virtual Java, o Eclipse é mais lento que softwares nativos. Para resolver as questões de velocidade a Google criou sua própria IDE de Android, chamada Android Studio. Pode ser que, no futuro, utilizar o Android Studio seja uma boa solução para os problemas de velocidade do Eclipse.

- **JUnit**

O JUnit foi escolhido como biblioteca de desenvolvimento de testes por ser a mais consagrada da linguagem Java. O JUnit possui como pontos positivos: facilidade de encontrar conteúdo de suporte online e simplicidade para integrar ao projeto no Eclipse. O maior ponto negativo do JUnit é a dificuldade de integra-lo ao sistema operacional do Android. Como os testes tem que funcionar utilizando o contexto Android, é essencial que sua integração se dê

de maneira fluida. Se faz necessário também utilizar classes que *Mockeiam* elementos do sistema operacional Android.

- **Wiremock**

A utilização da biblioteca Wiremock possui como pontos positivos: pouquíssimas linhas de código necessárias para resolver os problemas de mock e sintaxe extremamente coesa e simples. Como pontos negativos: dificuldade de encontrar documentação *online* e problemas para realizar a integração com o Android.

5.2.2: Outros aspectos

A modelagem do estado ideal do sistema foi útil para que fosse possível dar um contexto para os pedidos da empresa Camiolog. O processo CMMI utilizado mostrou quais são as outras melhores práticas que devem acompanhar a realização dos testes para a configuração de desenvolvimento, de fato, robusto. A Cheesecake Labs demonstrou que, no futuro, tem interesse em continuar melhorando os processos relacionados a testes (em conjunto com a Camiolog) baseando-se no CMMI.

A implementação do teste teve como pontos positivos: melhor compreensão de como funciona o *set-up* do ambiente de avaliação em sistemas operacionais embarcados (Android); velocidade altíssima de execução, devido à biblioteca de Mock da camada de rede; maior segurança no desenvolvimento de novas funcionalidades e, finalmente, confirmação do estado robusto da *feature* em questão.

A implementação do teste teve como pontos negativos: muita demora para integrar a rotina de avaliação com o SO Android e a existência de poucos recursos online, de suporte, disponíveis.

5.3: Análise das Conclusões dos Gestores

Os gestores da Cheesecake Labs deixaram claro que os esforços foram úteis para determinar qual deve ser o caminho trilhado no futuro em busca de processos de gestão mais controlados, organizados, documentados e profissionais. As mudanças realizadas nos projetos demonstram como pequenas iniciativas são capazes de alterar o fluxo de trabalho dos colaboradores de maneira imensa, direcionando os esforços no sentido correto, aumentando a satisfação dos clientes e, finalmente, fomentando a cultura de documentação e utilização de “melhores práticas” renomadas.

O corpo de gestão da Cheesecake Labs afirmou que o prazo definido para a realização das mudanças foi curto. A busca por melhorias nos processos de gestão não é algo que se conclui em poucos meses, devido a grande complexidade do sistema empresarial. Dessa maneira, fica claro que os problemas solucionados por este trabalho são apenas a “ponta do iceberg”, ou seja, os esforços deste são apenas o início de uma grande mudança que deve ser realizada de maneira gradual.

Com relação às melhorias produtivas nos processos gerenciais previamente citados, a equipe de gestores da Cheesecake Labs concluiu que o tempo de planejamento do projeto se tornou aproximadamente 50% maior e a previsão de redução do tempo de desenvolvimento do projeto é de 20% (devido aos esforços mais precisos e produtivos). É importante salientar que o tempo de desenvolvimento do projeto, na Cheesecake Labs, é de aproximadamente 70% da duração total e o tempo de planejamento é de aproximadamente 10%. Dessa maneira, pode-se calcular qual foi a variação aproximada no tempo total do projeto e, como consequência, a melhoria produtiva aproximada no sistema:

$$\text{Variação do tempo de total do projeto} = 10\% * 50\% - 70\% * 20\% = -9\%$$

Levando em conta os cálculos previamente citados, os gestores da Cheesecake Labs concluíram que a produtividade do projeto melhorou em aproximadamente 9%. Dessa maneira, melhorou-se o lucro do projeto, seu tempo de desenvolvimento e a capacidade de expansão da empresa.

Capítulo 6: Conclusões

Este capítulo tem como objetivo concluir o documento, explicando – de maneira breve – o resumo do trabalho, as conclusões críticas e os próximos passos a serem trilhados pela equipe de gerência da Chesecake Labs.

Em suma, o trabalho inicia-se com o cenário crítico de gerência da empresa Chesecake Labs. Para solucionar os problemas da empresa, é realizada uma pesquisa por metodologias de gestão de projetos e desenvolvimento de *software*. Após estudar as referências de maior destaque, são escolhidas as normas para guiarem dois estudos de caso com as empresas Cogentio e Camiolog.

Para aplicar as metodologias nos casos de estudo, foram feitos modelos dos processos atuais – pertencentes ao escopo do trabalho – considerados críticos pelas empresas parceiras. Em seguida, foram feitos modelos ideais, nas definições das normas escolhidas, das áreas de processos críticos. Em terceira instância, foram propostas e aplicadas melhorias para aproximar os modelos atuais do ideais.

Após a proposição e aplicação das melhorias, é feita uma análise crítica sobre o resultado das metodologias e as aplicações de mudanças nos casos de estudo.

As conclusões do documento podem ser divididas nos seus respectivos objetivos:

- As formas mais consagradas de gestão de projetos e processos de desenvolvimento de *software* foram estudadas de maneira satisfatória, pois diversos conceitos renomados foram utilizadas para a criação dos modelos ideais e, posteriormente, para a geração de melhorias. Dessa maneira, a equipe gerencial da Cheesecake se tornou mais íntima com os conceitos de gestão, fazendo com que esforços futuros se deem de maneira mais precisa e simples.
- Os processos críticos definidos pelos gestores da Cheesecake Labs foram modelados, analisados e foram propostas e implementadas melhorias nos dois casos de estudo. Dessa maneira, se fez possível compreender como as formas de gestão de projetos e processos de desenvolvimento de *software* renomadas são utilizadas em aplicações reais. O aprendizado prático foi de fundamental importância para a organização, pois em um cenário de negócios, é essencial ser capaz de trazer os conhecimentos teóricos para os processos vigentes.
- Os resultados das mudanças realizadas nos dois casos de estudo foram analisados, levando em conta a opinião dos gestores da Cheesecake Labs. A análise se mostrou útil para o planejamento de mudanças de médio prazo, pois explicita questões críticas que não foram resolvidos pelos esforços deste trabalho.

- As principais ferramentas da *web*, relacionadas aos temas do projeto, foram avaliadas preliminarmente e, em seguida, utilizadas para aprimorar os processos. O estudo e análise das ferramentas é de suma importância para os gestores da Cheesecake Labs, pois trazem à prática questões teóricas importantes. Mesmo que no futuro as ferramentas utilizadas mudem, o seu estudo foi essencial para que a equipe da Cheesecake Labs entenda quais funcionalidades existem no mercado e devem ser buscadas, quais os custos (em média) das ferramentas e, finalmente, quais os seus reais impactos na gestão da empresa.

Conclui-se que, no contexto de Engenharia de Controle e Automação (ECA), o trabalho trouxe aprendizados importantes. Os aprendizados são definidos como importantes pois o engenheiro de controle e automação deve ser capaz de: entender a problemática em questão e buscar, na literatura, ferramentas e metodologias para solucioná-la; Utilizando as ferramentas e metodologias, modelar o problema; Após modelado, criar soluções para geração de melhorias; Baseando-se nas melhorias, concluir sobre medidas futuras e, finalmente, reiniciar o processo (fechando a malha de atuação) utilizando os aprendizados previamente citados.

Bibliografia:

[1] – PMI, Project Management Institute, Um Guia do Conjunto de Conhecimentos em Gerenciamento de Projetos (Guia PMBOK®) Quinta Edição. Project Management Institute, 2013.

[2] – <http://www.pmi.org/>.

[3] – Zandhuis, A.; Stellingwerf R., ISO 21500 Pocket Guide. Van Haren, 2013 (disponível em: <http://www.vanharen.net/Samplefiles/9789087538095SMPL.pdf>).

[4] – Wagner R., ISO 21500 Guidance on Project Management clears final hurdle. GPM BLOG, 2012. (disponível em: <http://gpm-blog.de/iso-21500-guidance-on-project-management-nimmt-letzte-hurde/>).

[5] – Rehacek, P., Standards ISO 21500 and PMBoK® Guide for Project Management. International Journal of Engineering Science and Innovative Technology (IJESIT) Volume 3, Issue 1, January 2014 (disponível em: http://www.ijesit.com/Volume%203/Issue%201/IJESIT201401_36.pdf).

[6] – Gasik, S., Comparison of ISO 21500 and PMBOK® Guide. (disponível em: <http://www.sybena.pl/dokumenty/ISO-21500-and-PMBoK-Guide.pdf>).

[7] – Labriet, T. - Comparing PMBOK® Guide 4th Edition, PMBOK® Guide 5th Edition, and ISO 21500, STS SA, Lausanne, Switzerland, (disponível em: https://docs.google.com/file/d/0B_OLY9oI6F-EWmo4bTRWmjNpN1U/edit)

[8] – Chrissis, M. B.; Konrad, M.; Shrum S., CMMI®: Guidelines for Process Integration and Product Improvement, Addison-Wesley Professional, 2003 (disponível em: <http://my.safaribooksonline.com/0321154967>)

[9] – <http://whatis.cmmiinstitute.com/about-cmmi-institute>

[10] – CMMI Product Team, CMMI for Development, v 1.3. Carnegie Mellon, 2010.

[11] – <http://www.pontodatecnologia.com.br/2006/08/comparao-do-mpsbr-com-o-cmmi.html>

[12] – <http://www.bgtbrasil.com/bpmn.html>

[13] – Spínola R. O., Introdução à Construção de Diagrama de Classes Um Estudo de Caso, SQL Magazine 57 (disponível em: <http://www.devmedia.com.br/artigo-sql-magazine-57-introducao-a-construcao-de-diagrama-de-classes-um-estudo-de-caso/10861>).

[14] – Osbov, A., UML, Class and Sequence Diagrams. 2014 (disponível em: http://courses.cs.washington.edu/courses/cse403/12au/lectures/cse403_UML.pdf).

[15] – <http://www.idef.com/idef0.htm>

[16] – <https://www.lucidchart.com/>

[17] – <http://WBSTool.com>

[18] - <http://teamgantt.com>

[19] – <https://github.com>

[20] – <http://wiremock.org/>