

[AvalInsta]

VISÃO DO PROJETO

Versão [1.1]

Histórico de Revisão

Data	Versão	Descrição	Autor
16/02/2021	1.0	Detalhamento da Ideia principal do projeto.	Deivid, Felipe, Luiz, Victor
17/02/2021	1.1	Alteração da declaração dos objetivos, melhoramento da definição de atividades, melhoramento das metodologias de desenvolvimento e gerenciamento de riscos.	Deivid, Felipe, Luiz, Victor
10/03/2021	2.0	Detalhamento das sprints e o que será feito no 6.1.	Deivid, Felipe, Luiz, Victor
13/04/2021	3.0	Melhora de toda visão, preparando para entrega ao professor.	Deivid, Felipe, Luiz, Victor

Sumário

1	INTRODUÇÃO	4
1.1	Problema	4
1.2	Declaração do Problema	4
1.3	Objetivos do Projeto	4
2	STAKEHOLDERS	4
3	VISÃO GERAL DO PRODUTO	4
3.1	Declaração de Posição do Produto	4
3.2	Mínimo Produto Viável (MVP)	4
4	VISÃO GERAL DO PROJETO	4
4.1	Organização do Projeto	5
5	FERRAMENTAS, AMBIENTE E INFRAESTRUTURA	5
5.1	Hardware	5
5.2	Software	5
6	PROCESSO DE GERÊNCIA DE PROJETO	5
6.1	Planejamento das Fases e Iterações do Projeto	5
6.2	Processo de Desenvolvimento e Mensuração	5
6.3	Milestones e Objetivos do Projeto	5
6.4	Matriz de Comunicação	6
6.5	Escalabilidade do Projeto	6
6.6	Gerenciamento de Riscos	6
6.7	Critérios de Replanejamento	6
7	LIÇÕES APRENDIDAS	6
8	REFERÊNCIAS	7

VISÃO DO PROJETO

1 INTRODUÇÃO

1.1 Declaração do Problema

O problema	Falta de Feedback nos produtos vendidos pelo instagram
Afeta	Usuários do Instagram (lojas informais e usuário comum)
Cujo impacto é	Falta de confiança no pequeno empreendedor
Uma solução de sucesso seria	Criação de uma plataforma de avaliação e recomendação segura para compras realizadas no aplicativo.

1.2 Objetivos do Projeto

- Desenvolvimento de um site;
- Realização de um sistema de avaliação de lojas virtuais do aplicativo;
- Sistema de recomendação com base nas melhores avaliações cadastradas;
- Aumentar o alcance dos pequenos e microempreendedores digitais, pela divulgação de boas avaliações acerca dos seus Instagrams e verificado quando o usuário de nossa plataforma decide realizar uma compra e avaliação positiva por conta da nossa aplicação.
- Confiabilidade e diminuição de fraudes por parte de “vendedores digitais”. Medido pela quantidade de avaliações positivas e índice de recomendação do perfil. Quanto mais baixo o índice de recomendação, mais provável que o perfil não seja de confiança ou esteja realizando fraudes.

2 STAKEHOLDERS

<i>Nome</i>	<i>Descrição</i>	<i>Responsabilidades</i>
Alunos: Felipe Chermont, Deivid, Luiz, Victor	Estudantes de engenharia de software	Garante o desenvolvimento da aplicação web; Garante o monitoramento do progresso do projeto; Garante a resolução do problema proposto;
Usuários do Instagram	Pessoas que utilizam o aplicativo com interesse de realizar qualquer tipo de compra pelo mesmo	Fornecer feedbacks acerca da aplicação em termos de usabilidade e necessidade e como ela poderia ser melhorada.

Pequeno e Microempreendedor	empreendedores que visam realizar vendas através da plataforma do Instagram.	Fornecer feedback acerca de como a aplicação web proposta está afetando o negócio dele e como ela poderia ser melhorada.
-----------------------------	--	--

3 VISÃO GERAL DO PRODUTO

3.1 Declaração de Posição do Produto

Para	Usuários do instagram.
Quem	Usuários que pretendem realizar compras pelo aplicativo
O Avalista	É um site de avaliações online
Que	Recebe informações acerca de lojas virtuais e categoriza as avaliações. Promovendo maior alcance a lojas seguras, e alertando sobre possíveis fraudes.
Ao contrário	Da função de “Stories” do próprio vendedor, sujeito a fraude.
Nosso produto	Um site de avaliação de instagrams de venda

3.2 Mínimo Produto Viável (MVP)

- Validar se usuários do instagram se interessam na proposta de avaliar e recomendar os perfis em que realizaram alguma compra.

4 VISÃO GERAL DO PROJETO

4.1 Organização do Projeto

Papel	Atribuições	Responsável	Participantes
<i>Desenvolvedor full stack, Product Owner</i>	<i>Desenvolvimento do projeto, atualização do Backlog, Gerenciamento das histórias de usuário, organização do GitHub e site de documentação do projeto.</i>	<i>Felipe</i>	<i>Felipe, Deivid, Luiz, Victor</i>
<i>Desenvolvedor full stack, Scrum Master</i>	<i>Desenvolvimento do projeto, responsável pela organização das reuniões da equipe, dailys, sprints reviews e comprimento das tarefas pela equipe de desenvolvimento.</i>	<i>Deivid</i>	<i>Felipe, Deivid, Luiz, Victor</i>
<i>Desenvolvedor front end, arquiteto</i>	<i>Desenvolvimento do projeto, responsável pelo comprimento e organização da arquitetura do projeto.</i>	<i>Luiz</i>	<i>Felipe, Deivid, Luiz, Victor</i>
<i>Desenvolvedor front end, Designer</i>	<i>Desenvolvimento do projeto, responsável pelo design e identidade visual do projeto</i>	<i>Victor</i>	<i>Felipe, Deivid, Luiz, Victor</i>

<i>Orientador</i>	<i>Validação, orientação e auxílio no desenvolvimento do projeto.</i>	<i>George</i>	<i>George</i>
<i>Monitor</i>	<i>Auxiliar, encaminhar e verificar o andamento do projeto com o grupo de desenvolvedores</i>	<i>José</i>	<i>José, Leticia</i>

5 FERRAMENTAS, AMBIENTE E INFRAESTRUTURA

5.1 Hardware

<i>Perfil</i>	<i>Tipo de Hardware</i>	<i>Configurações</i>	<i>Qtd. Planejada</i>	<i>Prazo Estimado</i>	<i>Observação</i>
<i>Desenvolvedor</i>	<i>Computador</i>	<ul style="list-style-type: none"> <i>processador 4 cores/threads</i> <i>4 GB ram</i> 	<i>04</i>	<i>Já possuem</i>	

5.2 Software

<i>Perfil</i>	<i>Tipo de Software</i>	<i>Nome da Ferramenta</i>	<i>Versão</i>	<i>Qtd Licenças Planejada</i>	<i>Prazo Estimado</i>	<i>Observação</i>
<i>Desenvolvedor</i>	<i>Site/Aplicativo</i>	<i>GitHub</i>		<i>Não necessário</i>		<i>Armazenamento do projeto</i>
<i>Desenvolvedor</i>	<i>Aplicativo</i>	<i>Postgres</i>		<i>Não necessário</i>		<i>Banco de Dados</i>
<i>Desenvolvedor</i>	<i>Aplicativo</i>	<i>VSCode</i>		<i>Não necessário</i>		<i>Ferramenta para escrever códigos em python</i>
<i>Organizacional</i>	<i>Extensão GitHub/Site</i>	<i>ZenHub</i>		<i>Não necessário</i>		<i>Organização do Kanban Board</i>
<i>Comunicação</i>	<i>Aplicativo</i>	<i>Discord</i>		<i>Não necessário</i>		<i>Reuniões e comunicação da equipe.</i>
<i>Arquitetura</i>	<i>Aplicativo</i>	<i>Marvelapp</i>		<i>Não necessário</i>		<i>Protótipo de alta fidelidade</i>
<i>Arquitetura</i>	<i>Aplicativo</i>	<i>Balsamiq</i>		<i>Não necessário</i>		<i>Protótipo de baixa fidelidade</i>

6 PROCESSO DE GERÊNCIA DE PROJETO

6.1 Planejamento das Fases e Iterações do Projeto

Planejamento das fases do projeto				
Descrição	Ciclo 01		Data Início	Data Fim
Engenharia de Software	• Visão do Projeto (atualizado)		04/fev.	18/fev.
Descrição	Ciclo 02		Data Início	Data Fim
Requisitos de software	• Visão do Projeto (atualizado) • Backlog do Produto • Estrutura de requisitos de acordo com o SAFe • Quadro Kanban do Projeto (disponível para o professor) • Quadro Lean Inception (disponível para o professor)		23/fev.	11/mar.
Descrição	Ciclo 03		Data Início	Data Fim
Sprints				
Arquitetura e design de software	Sprint 01		16/mar.	20/mar.
	• Estudo das ferramentas • Estudo do conteúdo da Disciplina			
	Não será feito revisão com o monitor, por ser apenas o estudo das ferramentas	Revisão da equipe com todas as stacks do projeto definidas		
	Sprint 02		Data Início	Data Fim
	• Desenvolvimento do documento de arquitetura • Atualização do Backlog do produto • Criação do protótipo de baixa fidelidade		20/mar.	27/mar.
	Sprint Review/Sprint Planning/Reunião Monitor	Avaliação da Sprint, Planejamento da próxima sprint e Recomendações do monitor		

	Sprint 03		Data Início	Data Fim
	<ul style="list-style-type: none">• Criação do protótipo de alta fidelidade• Documento de Identidade Visual do Projeto<ul style="list-style-type: none">• Página de Cadastro• Página de registro de avaliação		27/mar.	3/abril.
	Sprint Review/Sprint Planning/Reunião Monitor	Avaliação da Sprint, Planejamento da próxima sprint e Recomendações do monitor		
Ciclo 04				
Construção de Software	Sprint 04		Data Início	Data Fim
	<ul style="list-style-type: none">• Página de Login• Refatoração da página de Cadastro• Refatoração da página de avaliação• Página de troca de senha		3/abril	10/abril
	Sprint Review/Sprint Planning/Reunião Monitor	Avaliação da Sprint, Planejamento da próxima sprint e Recomendações do monitor		
	Sprint 05		Data Início	Data Fim
	<ul style="list-style-type: none">• Atualizar o documento de visão do projeto<ul style="list-style-type: none">• Excluir uma avaliação• Edição de Avaliações• Ranquear outra Avaliação (like e dislike)• Atualizar pagina de Admin do Django		10/abril	17/abril
	Sprint Review/Sprint Planning/Reunião Monitor	Avaliação da Sprint, Planejamento da próxima sprint e Recomendações do monitor		
	Sprint 06		Data Início	Data Fim
	<ul style="list-style-type: none">• Recomendação de Lojas (baseado nas avaliações)<ul style="list-style-type: none">• Filtro de busca (baseado na categoria)• Excluir Avaliação pelo administrador (superUser)<ul style="list-style-type: none">• Comentário em avaliações		17/abril	24/abril
	Sprint Review/Sprint Planning/Reunião Monitor	Avaliação da Sprint, Planejamento da próxima		

		sprint e Recomendações do monitor		
MVP				
Ciclo 05				
	Sprint 07		Data Início	Data Fim
Testes	• Denúncia de Avaliações e Perfis • Testes (A ser estudado)		24/abril	1/maio
	Sprint Review/Sprint Planning/Reunião Monitor	Avaliação da Sprint, Planejamento da próxima sprint e Recomendações do monitor		
Entrega prevista do projeto funcional			01/05	

6.2 Processo de Desenvolvimento e Mensuração

- O projeto será elaborado utilizando algumas metodologias ágeis. Utilizaremos parte do Scrum modificado.
- Atribuído um papel para cada integrante, como “Product Owner”, Scrum Master e arquiteto de software. Além disso, todos serão desenvolvedores.
- Será feito Sprints mais curtas, com duração de uma semana, por conta do tempo curto para desenvolvimento do projeto.
- O backlog do produto será feito pelo P.O. juntamente a equipe de desenvolvimento.
- Atualização do que foi feito todos os dias através das dailys, ou seja, demonstrando através da plataforma escolhida (Discord) o que foi feito hoje, o que será feito no dia seguinte e quais foram as maiores dificuldades.

Será também utilizado o “Kanban board”, com os dados do BackLog, o que está sendo feito e o que já foi concluído, para assim podermos ter uma noção melhor de como está de fato progredindo o projeto.

- Ao final de cada ciclo será realizado as “Sprints Reviews”, os será analisado o “Kanban Board” e atualizaremos o BackLog conforme as necessidades da equipe e do

projeto. Será feito o uso do Planning Poker para consenso da equipe quanto às estimativas(tempo e esforço) dos itens do Product Backlog.

- Para melhorar a troca de conhecimentos e aprendizado da equipe, será utilizado o “Pair Programming” onde será feito o pareamento em duplas para troca de conhecimentos e desenvolvimento conjunto do projeto.

6.3 Matriz de Comunicação

<i>Descrição</i>	<i>Área/ Envolvidos</i>	<i>Periodicidade</i>	<i>Produtos Gerados</i>
<ul style="list-style-type: none"> - Acompanhamento das Atividades em Andamento - Acompanhamento dos Riscos, Compromissos, Ações Pendentes, Indicadores 	- Equipe de desenvolvimento	- Diário	-Descrição das atividades realizadas pelo grupo.
	- Equipe de desenvolvimento	- Semanal	-Análise da sprint e do BackLog do projeto.
-Verificar o andamento e progresso do grupo, auxiliando no que será realizado na próxima sprint;	-Monitor e equipe de desenvolvimento	- Semanal	-Avaliação das atividades exercidas, andamento geral do projeto e validação para desenvolvimento da próxima semana
- Comunicar situação do projeto	- Equipe de desenvolvimento	- Término de cada ciclo	- Ata de reunião e - Relatório de Situação do Projeto.
-Avaliação do projeto e recomendações para desenvolvimento futuro	- Equipe de desenvolvimento e professor Orientador	- Término de cada ciclo	-Avaliação do que foi realizado durante o ciclo e como será o andamento e ajustes para o próximo ciclo.

6.4 Escalabilidade do Projeto

Os conflitos mais comuns que podem aparecer durante o desenvolvimento do projeto são: Conflitos de prioridades, onde não existe um consenso de quais deveriam ser as prioridades das atividades do projeto. Conflitos de gerência, por conta de utilizarmos o método de alternância da gerência, algum membro pode não aceitar bem a gerência de outro colega de equipe. Conflitos de programação, sendo quatro membros distintos desenvolvendo o projeto, pode-se haver discordâncias na forma de se realizar alguma parte do projeto.

Todos os problemas são escaláveis para outras instâncias acima dos membros do projeto. Caso haja um conflito muito grande interno será preciso levar este problema ao professor.

Para evitar estes conflitos, utilizaremos o planejamento preventivo com o conjunto de técnicas descritas no guia PMBOK:

- Agrupamos diversos pontos de vista e opiniões acerca do problema e tomaremos a decisão com base no diálogo em conjunto;
- Encontraremos uma solução que traga a maior satisfação para todos os envolvidos no conflito;
- Suavizar o conflito e enfatizar os acordos acima das diferenças;
- Antecipar e evitar uma possível situação de conflito.

6.5 Gerenciamento de Riscos

Risco	Grau	Mitigação do Risco
Quebra do computador	Médio	Será distribuído o trabalho da sprint do participante para os demais integrantes do grupo.
Sobrecarga de trabalho	Alto	será feito uma reunião para analisar o problema e possíveis alterações.
Escolha errada das Stacks	Médio	Realização de uma refatoração do código.
Baixo conhecimento em programação nas Stacks escolhidas	Médio	A equipe se compromete em auxiliar todos os membros para aprendizado em conjunto.
Falta de experiência em gestão de projeto	Alto	Pode ser preciso alterar o processo de desenvolvimento, fixando a posição de liderança em alguém mais experiente.
Tempo insuficiente para entrega do projeto	Alto	Pelo tempo do semestre ser curto, pode ser preciso haver sobrecarga de trabalho ao final do projeto para entregá-lo no prazo.
Falta de comunicação da equipe	Alto	Pela equipe estar em diversos outros projetos e disciplinas pela UnB, a falta de comunicação e comprometimento podem acarretar na má elaboração do projeto, então é necessário a comunicação da equipe de forma direta e diária.

6.6 Critérios de Replanejamento

Quando percebido a inviabilidade de entregar o MVP no prazo da maneira como está sendo desenvolvido será realizada uma reunião para replanejamento dos métodos de desenvolvimento do projeto.

7 LIÇÕES APRENDIDAS

Aprendizados Unidade 1:

- Aprendemos inicialmente, o que é um projeto, o que é o gerenciamento de um projeto e como fazer este gerenciamento da forma correta utilizando o guia PMBOK.
- Aprendemos sobre as fases de desenvolvimento de software, onde é preciso seguir etapas respectivamente: Requisitos de software, design, construção, teste, manutenção, engenharia de coordenação, qualidade e engenharia de processos.
- Aprendemos sobre os métodos de desenvolvimento de software, desde métodos sequenciais lineares como o modelo cascata, até o desenvolvimento de metodologias ágeis como Scrum, XP, Lean, entre outras, seus princípios e quando é melhor utilizar cada metodologia. Como por exemplo, tendo as tecnologias e requisitos totalmente definidos, a metodologia clássica sequencial em cascata pode ser melhor que a Scrum por exemplo, pela quantidade de tempo que é gasto na metodologia ágil. Porém, quando é preciso ir analisando e reconfigurando o projeto ao longo de seu desenvolvimento o Scrum oferece um conjunto de soluções melhores que em cascata, mesmo envolvendo mais reuniões e organização.

Para o futuro, podemos melhorar alguns pontos que serão importantes para o desenvolvimento do projeto da melhor forma possível, sendo eles:

- Estudar mais acerca do gerenciamento e organização do projeto;
- Definir melhor e aprender a utilizar todas as plataformas e softwares que serão utilizados no desenvolvimento do projeto;

Aprendizados Unidade 2:

- Durante o desenvolvimento da segunda unidade da disciplina, aprendemos inicialmente sobre Engenharia de Requisito. Onde pudemos definir melhor “o que” será de fato realizado no projeto.
- Aprendemos sobre as etapas da engenharia de requisitos, sendo elas: elicitar, desenvolver, analisar, verificar, validar, comunicar, documentar e gerenciar os requisitos. e sobre a importância dessas etapas para o projeto, pois a qualidade do projeto depende essencialmente dos requisitos.
- Durante a unidade, aprendemos a realizar um “*Lean Inspection*” que de forma similar ao visão do projeto, nos obrigou a descrever melhor o projeto, realizando “*Brainstorming*” dos objetivos, pensando em como seria uma persona que utilizaria a nossa aplicação e sua jornada durante e como nosso aplicativo poderia contribuir com ela. O “*Brainstorming*” de funcionalidades que nos guiou durante o sequenciador, para entendermos de fato, o que será construído e o que estará disponível para o usuário da nossa aplicação realizar e qual será em média o esforço para o desenvolvimento dessas funcionalidades.
- O desenvolvimento do “*Lean Inspection*” também serviu como guia para o desenvolvimento do BackLog e das histórias de usuário. Outras atividades necessárias do projeto foram aprendidas e desenvolvidas ao longo da unidade.

- Além disso, foi aprendido sobre o Safe, sua estrutura baseada nos princípios Lean e Agile, e sobre os seus três níveis, time, programa e portfólio.

Para iniciar a etapa de codificação da aplicação, visamos aprender mais sobre as “*stacks*” que utilizaremos e melhorar a comunicação entre a equipe para conseguir seguir o planejamento proposto.

Aprendizados Unidade 3:

- Na unidade 3 foi visto sobre Arquitetura e design de software. Aprendemos sobre Design e mais especificamente design de software, que é uma forma de interpretar a implementação, analisando e compreendendo as interações entre determinadas partes do sistema.
- Dentro da parte de design, aprendemos sobre a arquitetura de software que seria uma maneira de ver a implementação de um produto/projeto, analisando os pontos que afetam o sistema inteiro e de todas as maneiras possíveis.
- A arquitetura do projeto pode ser diferenciada em diversos estilos, sendo especializações de elementos. aprendemos sobre os principais estilos, sendo eles, estruturas gerais, sistemas distribuídos, sistemas interativos, sistemas adaptáveis, etc. O que foi de extrema importância para o projeto, que utiliza o estilo de sistemas interativos com estilo MVC (model-view-controller).
- Estudamos mais especificamente este modelo para desenvolvimento no projeto, este modelo é separado em três partes, sendo elas:
 - Model: gerencia os dados do sistema;
 - View: define e gerencia como os dados são apresentados para o usuário;
 - Controller: Gerencia a interação do usuário;
- Aprendemos sobre modelos de Design, sendo o principal o UML (unified Modeling Language) que seria uma linguagem visual para especificar, construir e documentar os artefatos do sistema, e seus principais diagramas:
 - Diagrama de atividades;
 - Diagramas de casos de uso;
 - Diagramas de sequência;
 - Diagramas de classe;
 - Diagramas de estado;
- Com esse estudo, podemos modelar os principais diagramas do projeto e ter uma melhor visão de como deveria ser construída a nossa aplicação.
- Finalizando os conteúdos estudados na unidade de Design, foi visto o design de interface, sendo uma parte essencial do processo, que consiste na construção de uma visão geral de como será o software após sua construção, seguindo seus princípios (Aprendizagem, familiaridade do usuário, consistência, surpresa mínima, recuperabilidade, orientação do usuário e diversidade de usuário).
- Para desenvolvimento, estudamos mais a fundo a parte dos protótipos de interface, que auxiliam a equipe do projeto a identificar os requisitos de usabilidade, e ter uma espécie de guia para o desenvolvimento da parte de front-end do projeto. Os protótipos de Interface geralmente seguem três partes, protótipo de baixa fidelidade, protótipo de média fidelidade e protótipo de alta fidelidade, onde, o protótipo de baixa fidelidade seria uma ideia mais geral e rabiscada de como seria a aplicação. O de média fidelidade uma ideia mais bem implementada e campos mais bem elaborados, e o de alta fidelidade uma ideia mais fiel e certa de como deverá ser e se comportar o software no final de seu desenvolvimento.

- Para a etapa de codificação e testes, pretendemos ter todo esse conhecimento adquirido aplicado no projeto e todo o estudo das stacks escolhidas na parte da arquitetura concluída para colocar em prática a parte de construção do software.

Aprendizados Unidade 4:

- Para esta unidade, aprendemos como construir e desenvolver um software da melhor forma em termos de código. Aprendemos acerca da programação em pares, que consiste na programação realizada por duas pessoas para resolverem juntas o problema proposto em um mesmo computador. Esse método pode ser bastante eficaz, reduzindo a chance de erros, por estar sendo desenvolvido por duas pessoas e possuir revisão contínua de código.
- Aprendemos sobre o Clean Code que visa o bom desenvolvimento do código, e facilitar o trabalho de outros possíveis desenvolvedores que darão segmento com o trabalho realizado e segue alguns princípios importantes:
 - Elegante, fácil de se compreender e agradável de se ler;
 - Simples e direto;
 - Segue os princípios básicos de programação;
 - Sem duplicidade;
 - Bem testado;
 - Funções, métodos, parâmetros e variáveis com nomes significativos;
- Aprendemos sobre a importância da revisão de código, a revisão é importante para minimizar e prevenir de erros que podem atrapalhar o desenvolvimento do projeto, compartilhar o conhecimento e tudo que foi desenvolvido com o resto da equipe, além de aumentar a qualidade do software, sempre trabalhando em equipe.
- Outro conceito aprendido foi o de refatoração, onde é alterada a estrutura do código, sem alterar seu comportamento, permitindo melhoramento do design do código, seguindo os princípios do clean code e tornando o código melhor. Essa refatoração é realizada junto com testes, garantindo que o funcionamento do código estará se mantendo ao longo da refatoração.
- Por fim, aprendemos sobre a importância das soft skills, habilidades não técnicas que estão relacionadas a personalidade e inteligência emocional que auxiliam a pessoa a lidar melhor com o ambiente em que está sendo inserida. Como por exemplo: Comunicação, Empatia, Liderança, Colaboração, Proatividade, entre outras coisas. São muito importantes quando se trabalha em equipe e seu desenvolvimento é contínuo ao longo da vida.

8 REFERÊNCIAS

1	https://www.reclameaqui.com.br - Reclame aqui, contém reclamações de sites e lojas virtuais em geral.
---	--

2	https://www.tripadvisor.com.br/ - TripAdvisor, contém um sistema de recomendação e classificação de ambientes físicos e lojas virtuais de comidas.
3	https://g1.globo.com/economia/noticia/2019/05/23/golpistas-usam-instagram-para-aplicar-fraudes-em-vendas-on-line.ghtml - Através dessa notícia do G1 foi possível perceber a falta de uma plataforma específica de avaliação de lojas virtuais do Instagram.
4	Software Engineering Body of Knowledge swebok v 3.0 2013