# ESS 2020.2

Vinicius Garcia, Vinícius da Rosa Silva, fernando macedo, Ricardo Ferreira dos Santos Junior, Maria Estela, Renato Ferr...



# 1 mapa modular



### 30 ferramentas

0 Pontos [de convergência] 21 Pontos [de conversação]

# Distribuição na jornada



0

Indicadores

No período de 21/05/2021 até 25/08/2021 foram aplicadas 30 ferramentas na jornada, sendo 14 métodos de trabalho, 8 sistemas de tecnologia e 8 tópicos de aprendizagem.

Foram agendados 21 pontos [de conversação], sendo que 21 deles já foram fechados.

### **IF977**



No período de 21/05/2021 até 25/08/2021 foram aplicadas 30 ferramentas na jornada, sendo 14 métodos de trabalho, 8 sistemas de tecnologia e 8 tópicos de aprendizagem.

Foram agendados 21 pontos [de conversação], sendo que 21 deles já foram fechados.

# Engenharia de Software

A engenharia de software é a aplicação sistemática de abordagens de engenharia ao desenvolvimento de software.

Engenharia de software é uma disciplina de computação.

#### Questões essenciais

O que é Engenharia de Software e como ela se diferencia da programação?

O que faz um ótimo engenheiro de software?

Cite um grande caso de falha de software de larga escala (adicione como link neste kit) que trouxe imensos prejuízos (i.e. financeiros, pessoais, sociais, etc.) e diga o que faltou de Engenharia de Software que ocasionou a falha.

Hoje sabemos que as economias de TODAS as nações desenvolvidas são dependentes de software. Cada vez mais sistemas são controlados por software. Neste contexto, e sabendo que a Engenharia de Software se dedica às teorias, métodos e ferramentas para o desenvolvimento de software profissional envolvendo sistemas não-triviais e um conjunto de requisitos como guia, destaque um risco que pode contribuir para o fracasso de um projeto de software, sua probabilidade de ocorrer, como mitigar e qual ação tomar caso ele ocorra (ou contornar, ou resolver). Justifique sua resposta.

#### Referências

- Software Engineering Body of Knowledge (SWEBOK)
- Qual é o impacto da ES no mercado de Computação e na sociedade como um todo
- Engenharia de Software Moderna



### abordagem chuukese

Em um mundo global e turbulento, a implantação de novas tecnologias precisa seguir um modelo que enfrenta as intempéries conforme elas surgem – e não seguir um mapa previamente definido. Esta ferramenta tem por objetivo discutir o plano e a sua ausência em jornadas para alcançar objetivos

### Questões essenciais

Quais as similaridades você enxerga entre os desafios de mudanças tecnológicas para projetos de construção de software?

Em quais situações fariam sentido ter projetos que são executados dirigidos a um planejamento rígido e abrangente?

Quais informações podemos utilizar em um ciclo de vida de software para nos orientar em direção a garantir o atingimento dos nossos objetivos?

Qual o maior obstáculo para as abordagens dirigidas a plano em projetos de desenvolvimento de software? Como contorná-lo?

Cite um exemplo de uma mudança [antecipada] que pode ocorrer em projetos de desenvolvimento de software, e como tratá-la

Cite um exemplo de uma mudança [emergente] que pode ocorrer em projetos de desenvolvimento de software, e como tratá-la

Cite um exemplo de uma mudança [baseada na oportunidade] que pode ocorrer em projetos de desenvolvimento de software, e como tratá-la

### Referências

Gerindo a mudança tecnológica pelo improviso (PDF para alunos do Cln)

### Gerindo a mudança tecnológica pelo improviso (MIT Sloan)



Ouça o #4 episódio do Legal Talks no ar! PodCast da Queiroz Cavalcanti Advocacia, uma conversa sobre inovação, empreendedorismo e transformação digital.

### Questões essenciais

Nos últimos 10 anos mais que triplicou os números do empreendedorismo no país. A que devemos esse resultado, na sua opinião?

O que você classifica como empreendedor digital? E quais são os principais desafios no cenário brasileiro?

Quais são as principais competências do empreendedorismo digital e como você acha que a engenharia de software (em especial, nossa disciplina) vai ajudar a trabalhar essas competências?

O que determina um bom processo de inovação? Como ele é composto, formado?

Que atividades criativas não serão automatizadas por robôs e como se preparar para este novo cenário?

#### Referências

Apple

Spotify

Google



Constituição e demais informações das equipes

### Questões essenciais

Quais são so membros da sua equipe? Basta um dos membros informar a composição da mesma.

Já criou um repositório no github para a equipe? Informe aqui o link pra ele.

Quais os demais sistemas, serviços e/ou recursos que a equipe vai utilizar?

Tem alguma dúvida ou observação a fazer?

#### Referências



Um movimento que se sobrepõe à escola medieval e recupera os modelos de aprendizagem das escolas gregas, desta vez em escala global, viabilizado pelo mundo figital. Essencialmente sem salas de aula virtuais, simulacros

que já faliram no mundo físico e não serão mantidos no mundo figital, pois já não fazem sentido para toda essa geração pós revolução industrial, que não quer emprego, que não quer escola, que não quer mais fábricas, pelo menos não da forma como estão.

### Questões essenciais

Quais as principais diferenças entre a escola figital e a educação a distância?

Como a escola figital interfere no cenário contemporâneo de educação?

Além dessas questões, que outras você acrescentaria ao debate sobre escolas figitais?

#### Referências





### **Identificar Personas**

atividade prática para construção de personas

### Questões essenciais

A partir da observação do cenário, que grupos de pessoas frequentam o espaço?

Escolha um grupo de pessoas que frequenta o espaço e descreva uma persona que representa as pessoas desse grupo.

### Referências





### **Inferir Oportunidades**

atividade prática para identificação de oportunidades

### Questões essenciais

A partir da observação das personas quais comportamentos no presente podem impactar o negócio no futuro?

### Referências

oportunidades periféricas e críticas



### **Propor Hipóteses**

atividade prática para proposição de hipóteses

### Questões essenciais

Qual oportunidade você pretende tratar no projeto?

Que possíveis soluções você propõe para tratar a oportunidades?

### Referências

c<sup>2</sup> geração de alternativas



### Hackenge

Hackathon + Challenge = Método [de trabalho] cujo foco é criar um desafio prático para os participantes

#### Questões essenciais

Qual o desafio proposto?

Existem premissas, restrições ou condições?

Observações adicionais?

### Referências



Wiki | Hackathon



### Hackenge

Hackathon + Challenge = Método [de trabalho] cujo foco é criar um desafio prático para os participantes

### Questões essenciais

Qual o desafio proposto?

Existem premissas, restrições ou condições?

Observações adicionais?

### Referências



Wiki | Hackathon



### Automação de Testes

Metodologia de testes cujo foco é ter o desenvolvimento de software guiado por testes [unitários/funcionais + comportamentais] automatizados.

### Questões essenciais

O que diferencia as abordagens de TDD e BDD? Onde e como elas se conectam no ciclo de desenvolvimento guiado por testes?

O que são os princípios FIRST e quais os benefícios que ele pode trazer para qualidade de software?

Quais as diferenças (vantagens vs desvantagens) entre TDD e debugging (debugar)? Ter testes automatizados elimina a necessidade de um debugador? Justifique.

O que é um teste Flaky? Cite um motivo para esse comportamento.

O que é teste fim-a-fim? Por que testar fim a fim?

Qual a principal desvantagem do teste fim a fim na sua opinião?

#### Referências

- Engenharia de Software Moderna, Cap. 8: Testes
- Por que escrever testes automatizados se eu mesmo posso testar?
- Desenvolvimento Orientado a Testes
- Four reasons you need automated tests for your code
- How to add end to end tests to your project with Cypress



### **Web Frameworks**

Um web framework (WF) ou web application framework (WAF) é uma estrutura de software projetada para oferecer suporte ao desenvolvimento de aplicativos da web, incluindo serviços, recursos e APIs. Os WFs fornecem uma maneira padrão de construir e implantar aplicações na Web. Os WAFs visam automatizar a sobrecarga de trabalho associada às atividades comuns realizadas no desenvolvimento da web fornecendo bibliotecas para acesso a banco de dados, frameworks de templates e gerenciamento de sessão, reutilização de código e resolução de problemas comuns por meio de padrões de projeto.

### Questões essenciais

Como está o cenário de uso do padrão MVC nos dias atuais, com o crescimento do desenvolvimento orientado/guiado por APIs?

Qual a principal similaridade e diferença entre os padrões Model-View-Controller, Model-View-ViewModel e o Model-View-Presenter?

Seria possível classificar o uso de cada um deles, ou seja, onde cada um deles se destaca mais?

Onde entram novas tecnologias como Single Page Applications (SPA) e Progressive Web Applications (PWA) na "fila do pão" destes design paterns?

- Web framework | Wikipedia
- From MVC to Modern Web Frameworks
- Slides da aula Model-View-Controller; Olá AdonisJs!

- Slides da aula Criando CRUDI e relações em API REST no Adonis JS
- Slides da aula Finalizando MVC; Introdução a Behavior-Driven Design & User Stories
- 2 A Comparison of Architecture Presentation Patterns: MVC vs. MVP vs. MVVM
- Understanding The Difference Between MVC, MVP and MVVM Design Patterns
- ⊘ Model-View-Controller, Engenharia de Software Moderna, Cap. 7: Arquitetura Introdução ao Padrão MVC
- Aplicação de arquiteturas MVC e CA na prática | Podcast Faladev #27
- How the Model View Controller Architecture Works MVC Explained



Hackathon + Challenge = Método [de trabalho] cujo foco é criar um desafio prático para os participantes

### Questões essenciais

Qual o desafio proposto?

Existem premissas, restrições ou condições?

Observações adicionais?

### Referências

Wiki | Hackathon



Um protótipo é, acima de tudo, uma ferramenta de comunicação. É através do protótipo que informamos nossas ideias, apresentamos a terceiros nossas hipóteses e ouvimos suas críticas e sugestões.

### Questões essenciais

como será o protótipo?

o que precisa ser feito para produzir o protótipo?

quem participará do time de produção do protótipo?

quando o protótipo estará pronto para ser validado?

quais indicadores serão adotados para medir a performance do protótipo e quais as metas para cada indicador?

além dessas, que outras perguntas precisam ser respondidas antes de produzir o protótipo?

### protótipos incrementais e iterativos



### **Experimentar Protótipos**

É preciso experimentar, testar, validar hipóteses o tempo todo no mundo figital. A flexibilidade do digital permite o redesign e refactory dos produtos e serviços, promovendo uma espécie de amadurecimento continuado das experiências das pessoas.

### Questões essenciais

como o protótipo será testado?

quem vai aplicar o teste?

quem vai participar do teste?

quando o teste será aplicado?

onde o teste será aplicado?

o quê precisa ser perguntado no teste?

além dessas, que outras perguntas precisam ser respondidas antes de aplicar o teste do protótipo?

#### Referências



experimentos incrementais



### **Onboard IF977**

Método [de imersão] para identificar informações na construção de uma jornada SUFICIENTE para o estudante entender o contexto e EFICIENTE para ser usada como base para o processo de ENSINO e APRENDIZAGEM na disciplina

#### Questões essenciais

Qual sua EXPECTATIVA para esta jornada de aprendizagem?

Qual área dentro da computação te interessa atualmente (que você pretende se especializar, construir carreira)?

Você já se envolveu ou está envolvido em algum projeto de construção de software (aplicativo, aplicação, plataforma, sistema...)?

Quais PONTOS CRÍTICOS podem interferir no sucesso do seu aprendizado?

Qual CENÁRIO espera ver na sua formação depois desta disciplina?

#### Referências

Engenharia de Software (Bacharelado em Sistemas de Informação, CIn, UFPE)



As leis de evolução de software são mais conhecidas por Leis de Lehman e são dividida em oito leis, onde as três primeiras leis foram apresentadas em 1974, a quarta, quinta e sexta lei foram apresentadas em 1980 e as duas últimas leis em 1996

#### Questões essenciais

A Lei da Modificação Contínua (1974) versa que os softwares devem ser continuamente adaptados para não deixar de atender às necessidades para as quais foram criados. Já a Lei da Complexidade Crescente (1974) diz que à medida que o software sofre alterações, sua estrutura original passa a ser descaracterizada e sua complexidade aumenta, de modo que também aumenta gradativamente os custos de sua manutenção até o momento que a manutenção passe a se tornar inviável. Por fim, a Lei da Auto-Regulação (1974) por sua vez sugere que o software possui uma dinâmica própria definida, o que decide as tendências da manutenção, limitando o número de possíveis mudanças. Como lidar com estas 3 leis em um ciclo de vida de software?

Como você vê a relação e medição de forças entre o que diz a Lei da Conservação da Estabilidade Organizacional (1980), a Lei da Conservação da Familiaridade (1980) e a Lei do Crescimento Contínuo (1980)?

Quais as relações diretas e indiretas entre as consequências da Lei da Qualidade Declinante Modificação Contínua (1996) e da Lei da Realimentação do Sistema (1996)?

Com base no entendimento das 8 leis de Lehman, quero saber se a sua percepção sobre o que representa um caso de sucesso e de falha na engenharia de software mudou? O que representa uma falha épica, na sua opinião (justifique)?

#### Referências

- The Past, Present, and Future of Software Evolution
- On the Evolution of Lehman's Laws
- Lehman's laws of software evolution (Wikipedia)
- As 8 leis de Lehman foram o Manifesto do século XX



### mundo figital

O mundo físico ampliado pelo digital e orquestrado pelo social.

### Questões essenciais

Como o físico é ampliado pelo digital?

Como o físico é orquestrado pelo social?

Como o mercado muda a partir do figital?

Como você vislumbra o comportamento das pessoas, o consumo de produtos e serviços, e a experiência da jornada de clientes nesse futuro figital?

Além das questões acima, que novas questões você propõe sobre o tema para serem respondidas pelos demais participantes?

### O mundo figital



### **Escolher Cenários**

atividade prática para construção de cenários

#### Questões essenciais

Que cenário [ambiente e suas características geográficas, históricas e culturais] você propõe como ponto de partida para o projeto?

#### Referências





### trabalhos figitais

O trabalho mudou e muito desde a chegada da internet, mas foi agora, no último ano, que essa mudança se tornou mais evidente. A pandemia do COVID19 acelerou a mudança do trabalho como conhecíamos de físico para figital. Traballho remoto é apenas uma ponta desse movimento, quase que a casca sintática de uma transformação semântica que resultou em uma mudança pragmática nos mercados.

#### Questões essenciais

Como os métodos do design participativo mudam a criação no trabalho figital?

Como os métodos de desenvolvimento incremental interferem na produção de produtos no trabalho figital?

Como as metodologias ágeis impactam o trabalho figital?

Além das questões acima, que novas questões você propõe sobre o tema para serem respondidas pelos demais participantes?

### Referências





### tecnologias figitais

A base do ecossistema de bits que permite fazer a transição de negócios analógicos digitalizados para negócios digitais [ou figitais] está na convergência de quatro tecnologias: social, mobile, analytics, cloud. Estas, por sua vez, impulsionam boa parte da transformação digital que estamos vivenciando.

### Questões essenciais

Como tecnologias baseadas em redes sociais já influenciam e podem vir a influenciar o seu mercado?

Como tecnologias baseadas em mobilidade já influenciam e podem vir a influenciar o seu mercado?

Como tecnologias baseadas em computação em nuvem já influenciam e podem vir a influenciar o seu mercado?

Como tecnologias baseadas em analytics [big data] já influenciam e podem vir a influenciar o seu mercado?

Além das questões acima, que novas questões você propõe sobre o tema para serem respondidas pelos demais participantes?

#### Referências



Tecnologias habilitadoras



### Qualidade de Software

Muitos profissionais da nossa área têm dificuldades em definir qualidade de software devido à quantidade de aspectos que precisam ser considerados para avaliar se um software possui ou não qualidade, qual o nível de qualidade desejável, como medir qualidade e quais os obstáculos para obtê-la.

#### **Questões essenciais**

Como você define qualidade de software? Dê exemplos de software com qualidade e sem qualidade.

Qualidade, Qualidade de Software e Garantia da Qualidade de Software são as mesmas coisas?

Na sua opinião quais são os principais benefícios da gestão e controle de qualidade na Engenharia de Software?

#### Referências

- **Bate papo sobre o livro Engineering the Digital Transformation**
- Qualidade de software não é apenas funcionando
- Wiki | Qualidade de software
- **SWEBOK Guide V3 Topics**



### **Atributos de Qualidade**

Para um sistema de tecnologia ter sucesso, não basta apenas satisfazer os requisitos funcionais. Os sistemas críticos em geral devem atender também a segurança, confiabilidade, desempenho e outros requisitos semelhantes. Segundo a IEEE 1061, qualidade de software é o grau em que o software possui uma combinação desejada de atributos (por exemplo, confiabilidade, interoperabilidade)

### Questões essenciais

Quais são os três principais atributos de qualidade - do ponto de vista tecnológico - do sistema de tecnologia (exemplo de atributos de qualidade: mobilidade, segurança, desempenho, usabilidade, confiabilidade)?

Quais sao os três principais recursos do sistema de tecnologia? O que estes recursos entregam de valor?

Em se tratando de um sistema que já existe uma versão (mesmo que especificação ou projeto), quais recursos existem no sistema de tecnologia, mas não deveriam existir no momento (não agrega valor, não faz sentido, ou qualquer outro motivo)?

iverer entoids

- © Os atributos de qualidade na ISO/IEC 9126
- Quality attributes in Software Architecture
- Software Product Quality [ISO/IEC 25010]

# Teste de Software

Mudança de software é inevitável. Novos requisitos surgem quando o software é usado, o ambiente de negócio muda, erros devem ser reparados, novos computadores e equipamentos são adicionados ao sistema, o desempenho ou a confiabilidade do sistema deve ser melhorada. Um problema-chave para as organizações é a implementação e o gerenciamento de mudanças em seus sistemas com garantia de qualidade e aderência às necessidades dos clientes. Uma ferramenta de apoio a isso são os Testes de Software.

#### **Questões essenciais**

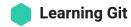
Conceitualmente, existe uma classificação para ERRO, FALHA e FALTA. Quais técnicas de Teste de Software podem ajudar a mitigar, capturar e tratar cada um deles, no contexto de VALIDAÇÃO e VERIFICAÇÃO?

Segundo Amman (2008), existem os seguintes níveis de Teste de Software: Teste Unitário, Teste de Módulo, Teste de Integração, Teste de Sistemas, Teste de Aceitação, Alpha/Betha testing. Quais as principais técnicas/ferramentas de Teste de Software que podemos utilizar para melhorar nossa cobertura considerando os diferentes níveis de teste?

Somente testes exaustivos podem mostrar que um programa está livre de defeitos. Contudo, testes exaustivos são impossíveis! Definir políticas de teste ajudam a estabelecer uma abordagem a ser usada na seleção de testes de sistema. O que devemos levar em conta na definição de políticas de teste?

#### Referências

- 2 TechTalk: Como aprendi que testar software é importante, Prof. Dr. Maurício Aniche
- What is Software Testing? The 10 Most Common Types of Tests Developers Use in Projects
- Engenharia de Software Moderna | Testes
- Introduction to Software Testing



Sequência de atividades para se familiarizar com o sistema de controle de versão Git

### Questões essenciais

Faça login no [GitHub] (primeira referência). Crie um repo privado chamado <LOGIN>-if977 (por exemplo, no meu caso o repo se chamaria VCG-IF977). Vá para Configurações, Colaboradores e Equipes e adicione os monitores e o professor como um colaborador.

Resolva os primeiros quatro níveis da aplicação [Learn Git Branching] (segunda referência): Sequência introdutória, Acelerando, Movendo trabalho por aí, Sortidos. Para obter crédito extra, complete Temas avançados. Para o envio, você só precisa demonstrar a conclusão dos níveis, o que pode ser feito por meio de uma captura de tela. No

problemas no futuro ou como se recuperar caso seu progresso seja perdido.

Crie um repositório git local (usando git init) em um novo diretório. Crie um arquivo "post-commit" em .git/hooks/. Dentro do arquivo, crie um comando que abrirá uma página da web imediatamente após um commit ser executado para aquele repo. Dica: start para windows, open para mac/linux. Em sua solução, forneça o conteúdo de "post-commit". Por fim, faça um screencast ou uma gravação GIF do processo.

#### Referências



- Learn Git Branching
- Why do you need to put #!/bin/bash at the beginning of a script file?
- LICEcap simple animated screen capture tool for Windows and OS X



Hackathon + Challenge = Método [de trabalho] cujo foco é criar um desafio prático para os participantes

### Questões essenciais

Qual o desafio proposto?

Existem premissas, restrições ou condições?

Observações adicionais?

### Referências

Wiki | Hackathon



contar histórias de como as hipóteses propostas impactam a vida das pessoas

### Questões essenciais

Era uma vez [como é fisicamente e onde vive o personagem principal da história?]

Que todos os dias [qual a rotina do personagem principal da história antes de conhecer a nossa hipótese?]

Até que um dia [como o personagem principal da história tomará conhecimento daquilo que estamos propondo?]

E por causa disso [quais as ações realizadas pelo personagem principal da história a partir do contato com o que estamos propondo?]

E desde então [quais os impactos positivos na rotina do personagem principal da história depois de ter contato com o que estamos propondo?]



Hackathon + Challenge = Método [de trabalho] cujo foco é criar um desafio prático para os participantes

### Questões essenciais

Qual o desafio proposto?

Existem premissas, restrições ou condições?

Observações adicionais?

#### Referências



Wiki | Hackathon



### **Melhorar Protótipos**

É preciso estabelecer um fluxo contínuo de mudanças, de pequenas melhorias da experiência de uso, da tecnologia e dos modelos de negócio articuladas no tempo, com tempo para amadurecer e causar mudanças no próprio negócio e no ecossistema como um todo.

### Questões essenciais

As pessoas entendem como funciona o protótipo?

As pessoas gostam do protótipo?

As pessoas afirmam ter orgulho de usar o protótipo?

As pessoas indicariam o protótipo para outras pessoas?

O que precisa ser modificado do ponto de vista da experiência do uso?

O que precisa ser modificado do ponto de vista da tecnologia?

O que precisa ser modificado do ponto de vista do modelo de negócio?

Além das questões acima, que novas questões você propõe sobre o tema para serem respondidas pelos demais participantes?

### Referências



melhorias incrementais e iterativas



Método [de trabalho] para apresentação da ideia do negócio

### Ouestões essenciais

#### questoes essentials

Persona: qual o perfil dos possíveis clientes do negócio?

Oportunidade: em qual problema da persona o negócio vai atuar?

Solução: qual a solução para o problema o time desenvolveu?

Diferenciais: qual o diferencial da solução em comparação a outras que já existem?

Negócio: como o time irá ganhar dinheiro com a solução (modelo de negócio)?

Time: quem são os membros do time e quais as funções de cada um no negócio?

### Referências

⊘ Como fazer uma apresentação direta e curta

Uma maneira rápida e estruturada para apresentar seu negócio