

Apresentação do Curso ICC410 – Prática de Análise e Projeto de Sistemas

1

Professor: D.Sc. Arilo Claudio Dias Neto

arilo@icomp.ufam.edu.br



Por que uma disciplina de prática?



Você se deixaria ser operado por um médico sem experiência prática?



Você viajaria num avião comercial com um piloto que não tem nenhuma hora de voo?



Você confiaria num edifício de 100 andares feito por um engenheiro que nunca construiu nada real?

Mas você não pergunta do médico, do piloto e nem do engenheiro a experiência prévia deles. Quem faz isso?

2

2

Por que uma disciplina de prática?

Então porque um engenheiro de software sem experiência poderia desenvolver um sistema importante para uma empresa?



Engenheiro de Software



3

3

Objetivo desta disciplina

Vivenciar a experiência de um projeto real tentando simular suas características e dificuldades

- Para isso:
 - Engenharemos um projeto real.
 - Trataremos com clientes reais.
- Utilizaremos metodologia e técnicas reais.
- Os prazos e comprometimento com eles serão reais.
- Apenas 1 coisa não será real: não receberão dinheiro por ele.
 - Mas suas notas dependerão exclusivamente do sucesso do projeto.



4

4

Relacionamento de ICC410 com outras disciplinas

Relacionamento \neq Dependência

Objetivo: ao pensar num trabalho prático, adotar um já em andamento



5

5

O que é Engenharia de Software?

"Engenharia de Software é a aplicação de uma abordagem sistemática, disciplinada e quantificável ao desenvolvimento, operação e manutenção de software"

IEEE Std 610.12 (1990)

6

6

Engenharia de Software

- Atividade realizada em grupo
 - Se estou sozinho, não preciso documentar, comentar, revisar algo que eu mesmo fiz
 - Se estou sozinho, minha capacidade de desenvolver algo mais complexo se restringe proporcionalmente ao tempo da tarefa
- Diferentes papéis participam desse processo
 - Desenvolvedor, projetista, gerente, testador, analista, cliente, usuário, dentre outros.



7

7

Como funcionará este curso?

- Requisitos para participar do curso
 - Vontade de aprender ; Dedicção ; Comprometimento com o curso e seus colegas ; Disciplina ; Organização.
- Grupos formados na primeira semana até o fim do curso
 - 4 pessoas
- Cada grupo ficará com 1 projeto de software real e deverá desenvolver este projeto desde sua concepção até sua entrega (após ser testado) ao usuário final.
 - Isso significa que o fim do período ficará apertado?
 - ERRADO: adotaremos uma metodologia de desenvolvimento iterativa!



8

8

Como funcionará este curso?

- Metodologia de Desenvolvimento adotada:
 - **Scrum:** metodologia ágil dentre as mais utilizadas na indústria.
 - **Iterações (Sprints):** 3 semanas (no mundo real em geral são 2 semanas)
- Papéis a serem alocados:
 - **Product Owner:** responsável por fornecer os requisitos para o projeto.
 - **Scrum Master:** responsável pelo gerenciamento da sprint.
 - **Developers:** responsável pelo desenvolvimento e testes unitários.
 - **Tester:** responsável pelos testes funcionais de aceitação.



9

Scrum Master

- Função:
 - Liderança da equipe.
 - Remove impedimentos das equipe para atingir um objetivo e desenvolver as entregas de um *sprint*.
 - Planejar as atividades do sprint
- Responsabilidades no Projeto:
 - Coordenar as reuniões (*daily meetings*)
 - Distribuir tarefas entre a equipe
 - Garantir o cumprimento dos prazos
 - Auxiliar o desenvolvimento
 - Cumprir as histórias do usuário (e seus critérios de aceitação)



10

Developers 1 e 2

- Função:
 - Desenvolvimento das histórias seleccionadas para o sprint.
 - Avaliação do código desenvolvido antes de liberar para testes.
 - Correção dos problemas detectados pelo *tester*.
- Responsabilidades no Projeto:
 - Criação de código que funcione
 - funcional e não funcionalmente
 - Participar dos *daily meetings*
 - Garantir o cumprimento dos prazos
 - Criação do BD e suas conexões com o código



11

11

Tester

- Função:
 - Planejamento e Execução dos Testes Funcionais.
 - Reporte dos problemas detectados aos *developers*.
- Responsabilidades no Projeto:
 - Criação dos casos e procedimentos de testes funcionais
 - Auxílio na definição dos critérios de aceitação das histórias
 - Execução dos testes criados
 - Garantir a qualidade do produto final



12

12

Product Owner (PO)

- Funções:
 - Auxiliar as equipes
 - Substituir o cliente
 - Apoio às tecnologias (método ágil, Scrum, framework de desenvolvimento, BD)
- Responsabilidades:
 - Não irá desenvolver nenhuma funcionalidade
 - Avaliar os projetos ao final dos sprints
- Perfil:
 - Super calmo e paciente ;)



Arilo Claudio
Professor da
Disciplina ICC410

13

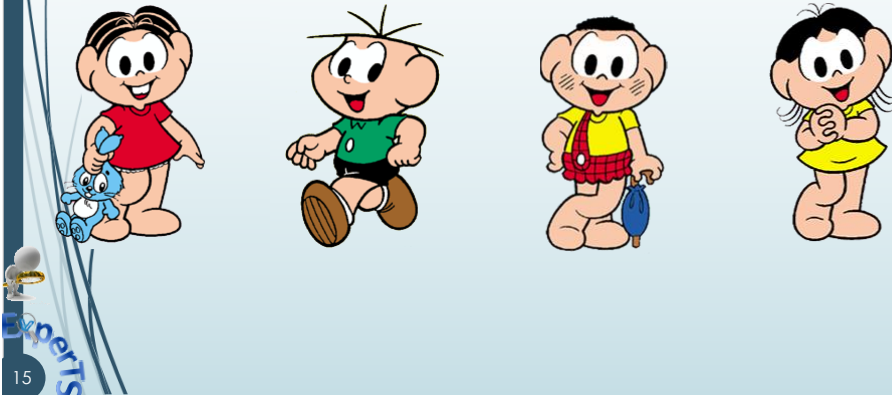
13

Quem assumirá cada papel? Posso escolher?

NÃO! Essa é a parte mais divertida do curso!

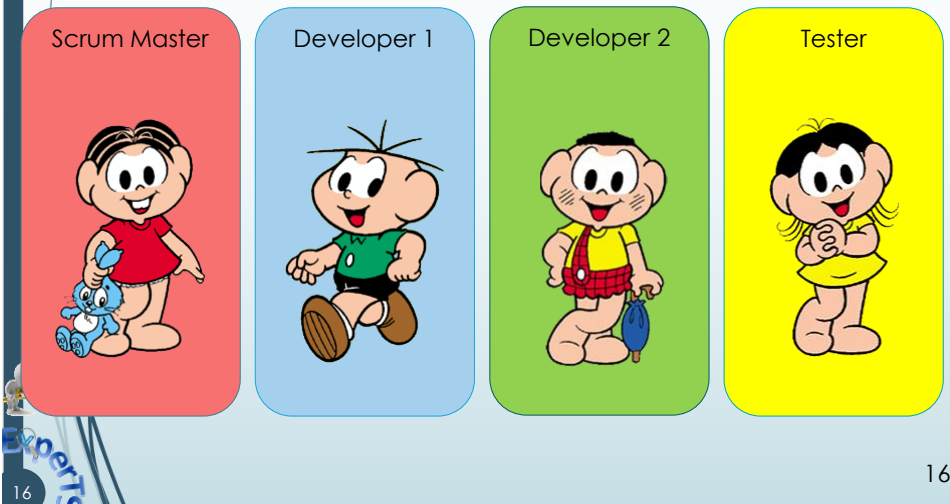
Como funcionará este curso?

- Imaginem uma equipe formada pelos 4 abaixo:



Como funcionará este curso?

- Iteração 1 (Sprint 1) :



Como funcionará este curso?

- Iteração 2 (Sprint 2) :

Scrum Master



Developer 1



Developer 2



Tester



17

17

Como funcionará este curso?

- Iteração 3 (Sprint 3) :

Scrum Master



Developer 1



Developer 2



Tester

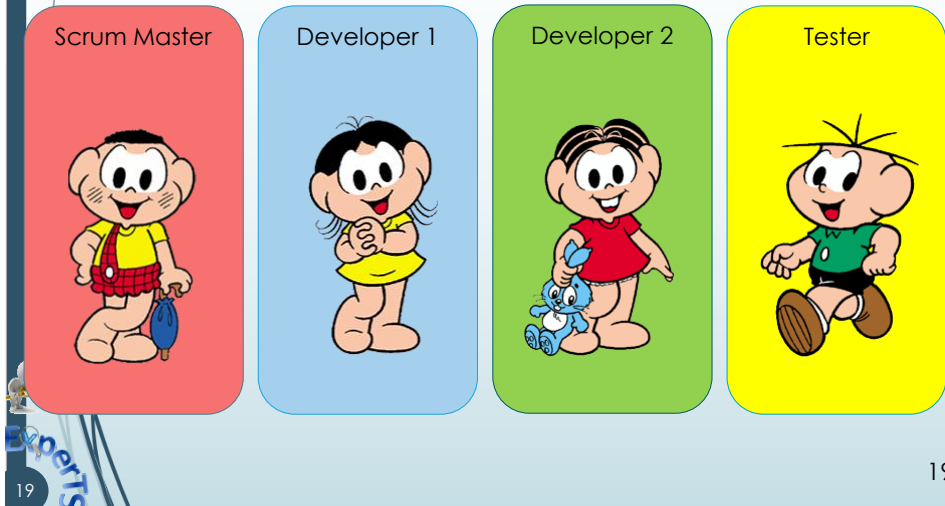


18

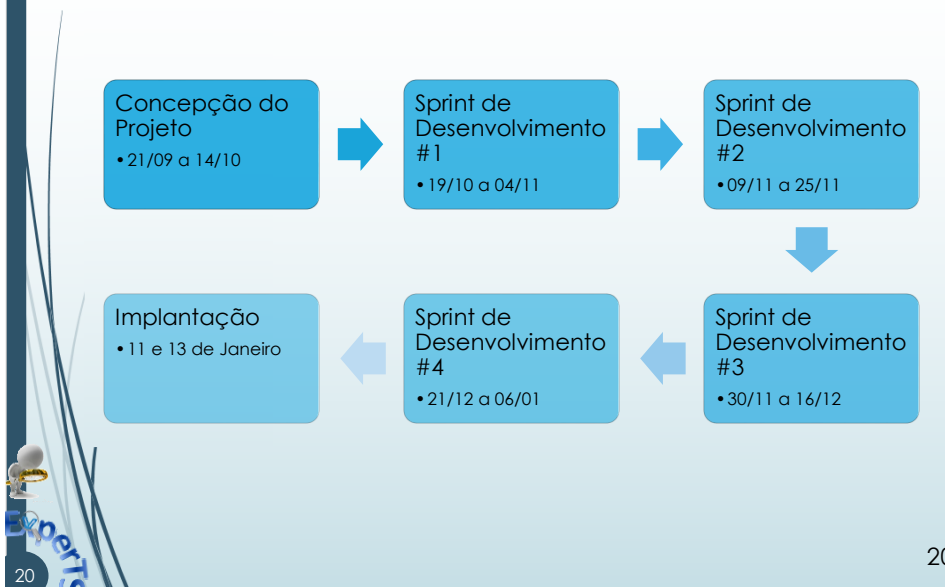
18

Como funcionará este curso?

- Iteração 4 (Sprint 4) :



Quais as etapas deste curso?



Etapa: Concepção do Projeto

- Datas:
 - 21 a 30 de Setembro: Entrevistas com Usuários/Clientes
 - 05 a 14 de Outubro: Modelagem, Prototipação e Apresentação
- Responsabilidade: **TODOS!!!**
- O que deve ser realizado:
 - Elicitação de requisitos
 - Configuração do ambiente de desenvolvimento
 - GitHub: versionador de código
 - Ambiente de Desenvolvimento (Yii + MySQL)
 - Trello: gerenciador de tasks no Scrum
 - Ferramenta online para descrição das estórias (ex: wiki)
 - Ferramenta online para registro dos daily meetings (ex: wiki)
 - Ferramenta para modelagem/prototipação do sistema
 - Modelagem dos Casos de Uso/Estórias elicidadas.
 - Modelagem do BD (Disciplina de Prática de BD).
 - Plano de entregas do projeto (por Sprint)
 - MVP: Minimum Value Product
 - Cronograma de atividades do projeto (desenvolvimento e teste)



21

21

Etapa: Sprints de Desenvolvimento



O que deve fazer:

- Coordenar os Daily Meetings
- Cobrar registros das atividades
- Gerenciar o Scrum Board
- Montar o Burndown Chart
- Resolver impedimentos
- Desenvolver Código



O que deve fazer:

- Participar dos Daily Meetings
- Desenvolver Código
- Realizar testes de unidade
- Assegurar a qualidade do seu código



O que deve fazer:

- Participar dos Daily Meetings
- Planejar, projetar e executar testes funcionais
- Nível de automação dos testes deve ser de no mínimo 50%
- Assegurar a qualidade do código final
- Montar o manual de utilização do sistema



22

22

Etapa: Sprints de Desenvolvimento

- O que deve ser entregue ao final de cada Sprint (via github ou wiki):
 - Código Fonte da Aplicação Funcionando.
 - Banco de Dados da Aplicação funcionando.
 - Casos e Scripts de Testes Funcionais.
 - Laudo de Execução dos Testes.
 - *Scrum Board* (Quadro Scrum).
 - *Burndown Chart* (Gráfico Burndown).
 - Manual do Usuário.
 - Planejamento da próxima Sprint (lista do que foi ou não cumprido, lista do que vai ser construído na próxima fase)

23

23

Etapa: Implantação

- O que deve ser feito/entregue:
 - Implantação do Sistema em Ambiente de Produção (ex: servidor do IComp)
 - Apresentação do projeto pronto ao cliente.
 - Faremos uma prévia ao final do Sprint 2 para eventuais ajustes de escopo.
 - Entrega da Documentação do sistema:
 - Estórias/Casos de Uso
 - Casos de Teste
 - Manual do usuário.
- **Requisito mínimo para a aprovação na disciplina.**

24

24

Scrum Board



25

Burndown Chart

Neste eixo vertical é mostrado a quantidade de trabalho a ser completado.



As datas ou dias de execução ficam neste eixo

26

Não teremos mini-provas, certo?



QUASE ISSO....

27

Daily Meeting nas Aulas (Notas)

- Coordenado pelo Scrum Master...
- Cada equipe fará um *daily meeting* (registrado) em cada aula:
 - O que cada um fez desde a última aula?
 - O que cada um fará até a próxima aula?
 - Algum impedimento para alguém não ter feito suas tarefas?



28

Daily Meeting nas Aulas (Notas)

- Cada aluno receberá uma nota pelo resultado do *Daily Meeting* (como recebia nas mini-provas)
- A ser realizado nos **primeiros 15 minutos** de cada aula
- Ausência implica em 0 (ZERO) na nota (dos ausente)
- Registrar A CADA AULA o resultado do Daily Meeting na Wiki do Projeto



29

29

Sprints

Início	Fim	Categoria	Tópico
14/set	30/set	Aulas	Apresentação da Disciplina e Elicitação de Requisitos
05/out	14/out	Especificação	Modelagem dos Requisitos e Prototipação
19/out	04/nov	Sprint 1	Desenvolvimento da Sprint 1
09/nov	25/nov	Sprint 2	Desenvolvimento da Sprint 2
30/nov	16/dez	Sprint 3	Desenvolvimento da Sprint 3
21/dez	06/jan	Sprint 4	Desenvolvimento da Sprint 4 (Ajustes)
11/jan	13/jan	Entregas	Implantação/Entrega dos Projetos

30

30

Tecnologias a serem adotadas

- Metodologia de Desenvolvimento: Scrum
- Ferramenta de Gestão do Projeto
 - Sugestão: Trello ou Kanban ou Taiga
- Ferramenta de Descrição e Prototipação de Estórias
- Linguagem de Programação e SGBD
 - Maioria dos projetos: PHP e MySQL (Yii)
- Versionador de código: GitHub
- Postagem de Estórias e Daily Meetings: Wiki
- Ambiente de desenvolvimento
 - Eclipse, Notepad++ ou qualquer outro



31

31

Avaliação

- Média Parcial = $(N1 + N2 + N3 + N4 + N5) / 5$
 - N1 = Nota da Especificação dos Requisitos
 - N2 = Nota como *Developer 1*
 - N3 = Nota como *Developer 2*
 - N4 = Nota como *Tester*
 - N5 = Nota como *Scrum Master*
- Média Final = $(2 \times \text{Média Parcial} + \text{Nota Final}) / 3$
 - Nota Final = Nota Final dada ao Projeto da Equipe
 - OBS: as notas são individuais, mas refletem o desempenho do aluno NO GRUPO.
- **APROVADO**
 - (Presença $\geq 75\%$) E (Média Final ≥ 5)
- **REPROVADO**
 - Caso contrário



32

32

Avaliação

- Avaliada no dia da *release* (entrega)
 - Analisando-se as notas dos daily meetings
- Divulgadas apenas AO FINAL do curso
- Critérios:
 - Nota da Especificação dos Requisitos
 - Não entregou TODOS os requisitos especificados
 - Nota máxima < 5,0 (podendo ser menor)
 - Entregou TODOS os requisitos especificados
 - Nota máxima = 10,0 (podendo ser menor)
 - Nota dos Sprints
 - Não entregou TODOS os *products backlog* (incluindo código + testes + manual) planejados
 - Nota máxima < 5,0 (podendo ser menor)
 - Entregou TODOS os *products baklogs*
 - Nota máxima = 10,0 (podendo ser menor)

33

33

Importante!!!

- Presença:
 - [Regulamento da UFAM](#)
 - Aluno deve ter o mínimo de **75%** de presença para ser aprovado (Cap. VI, Art. 77).
 - Nenhuma falta será abonada (Cap. VI, Art. 77).
 - Será feita chamada antes do início da aula e após seu término (2 presenças por dia).
- Observação:
 - **Burlar a regra do curso durante a aula ou fora do horário de aula implicará em punição ao aluno e/ou grupo.**
 - (ex: plágio ou executar um papel que não é seu naquela sprint)



34

34

Primeiras orientações!!

1. Definam as equipes agora!!!
2. Definam o personagem de cada membro da equipe (Mônica, Cebolinha, Cascão ou Magali).
3. Planejem suas atividades de elicitación de requisitos
4. Para 1 semana
 1. Preparem o ambiente que irão adotar
 2. Preparem o ambiente de repositório online de código
 3. Preparem uma ferramenta de gestão
 4. Preparem uma ferramenta de modelagem e prototipação
 5. Preparem uma máquina para trabalharem na aula

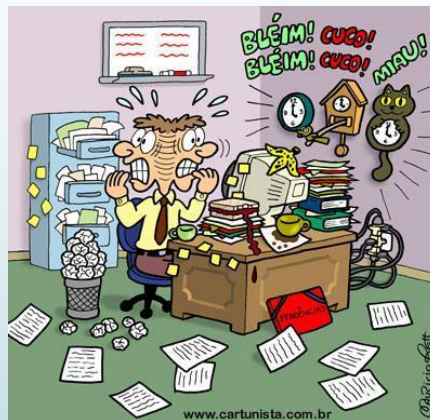


35

35

Conselho às Equipes!!!

- **CONSELHO 1:** Organizem-se para iniciar o projeto ATÉ a próxima semana, pois a primeira sprint começa início de Outubro, mas qualquer adiantamento (ambiente, tecnologia, estratégia de equipe) ajudará bastante!!!



36

36

Conselho às Equipes!!!

- **CONSELHO 2)** Usem os horários das aulas (cuja presença é obrigatória) para tirar dúvidas sobre Engenharia de Software, sobre as tecnologias adotadas e sobre o sistema.



37

37

Conselho às Equipes!!!

- **CONSELHO 3)** Usem as tecnologias para ajudar vocês, não para atrapalhar (monitoramento de tarefas, comunicação)



38

38

Conselho às Equipes!!!

- **CONSELHO 4:** Planeje bem seus sprints!!! Isso implicará no que vocês conseguirão cumprir ou não (consequentemente em suas notas) e principalmente nos próximos sprints (precisa balancear a carga de trabalho).



39

39

Conselho às Equipes!!!

- **CONSELHO 5:** Acompanhem diariamente seus sprints!!! Um projeto não atrasa um mês em uma noite, mas sim com os atrasos do dia-a-dia.

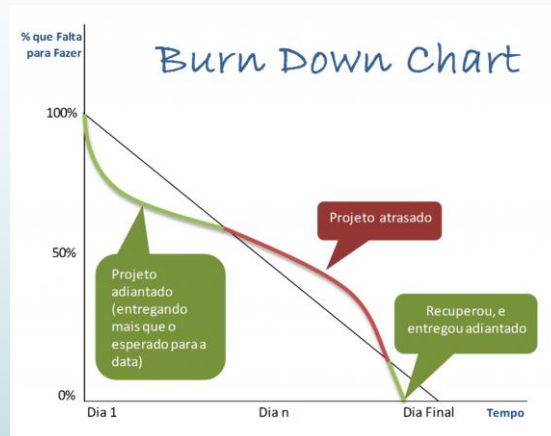


40

40

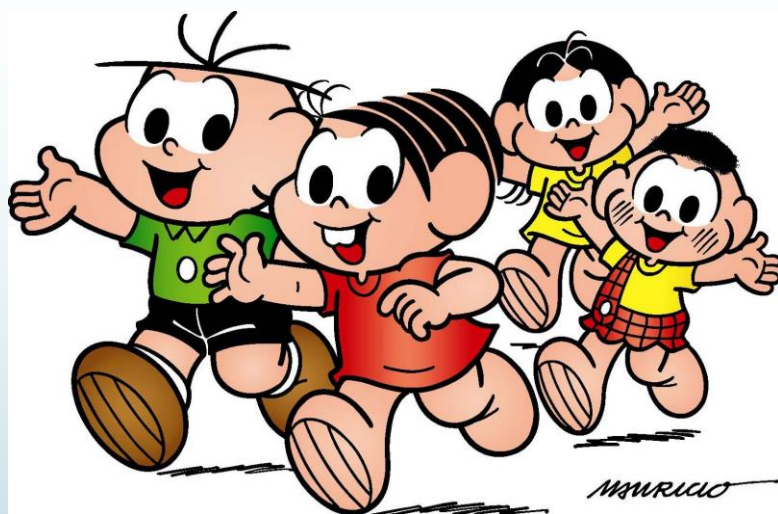
Conselho às Equipes!!!

- **CONSELHO 6:** Intervenham imediatamente ao encontrar problemas em seus sprints!!! Não deixe que isso se propague dia-a-dia, prejudicando o projeto inteiro!!!



41

Esperamos que todos fiquem assim!!!



42

Apresentação do Curso ICC410 – Prática de Análise e Projeto de Sistemas

43

Professor: D.Sc. Arilo Claudio Dias Neto

arilo@icomp.ufam.edu.br

