

# Modelação e Design

## 26 e 27: Processos de Desenvolvimento de software

Leonor Melo

leonor@isec.pt

1

### Construir software

- Atividades desenvolvidas durante a construção do software:
  - Definição do problema
  - Desenvolvimento dos requisitos
  - Planeamento da construção
  - Arquitetura do software ou visão de alto-nível
  - Design detalhado
  - Codificação e *debugging*
  - Testes unitários
  - Teste de integração
  - Integração
  - Teste de sistema
  - Correções de manutenção

Leonor Melo

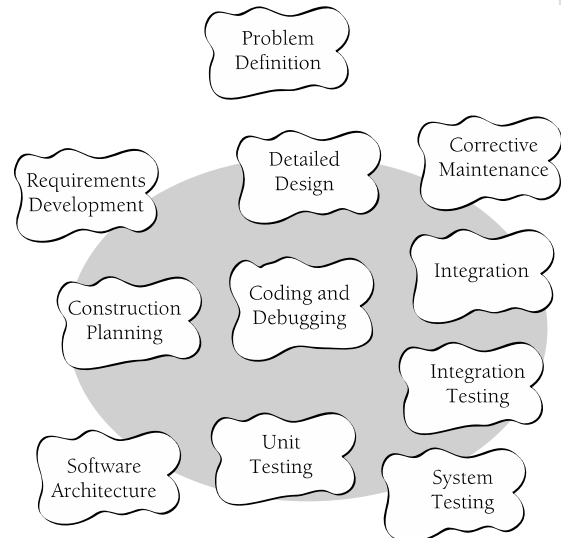
26 e 27 Processos de desenvolvimento de Software

2

2

## Construir software

- Manter em mente as diversas atividades permite criar uma perspectiva mais rica do que deve ser feito e como



Leonor Melo

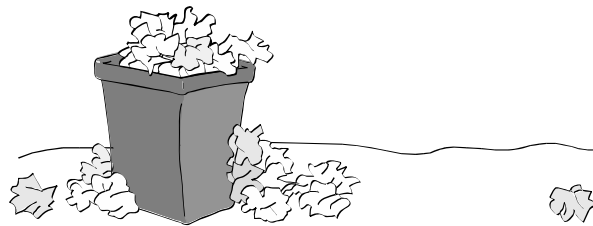
26 e 27 Processos de desenvolvimento de Software

3

3

## Construir software

- Desenvolver o sistema como quem escreve uma carta:
  - Sentar ao teclado e começar a escrever do início ao fim
    - Só funciona para projetos muito pequenos desenvolvidos por uma única pessoa
  - Processo "tentativa e erro"



Leonor Melo

26 e 27 Processos de desenvolvimento de Software

4

4

## Construir software

- Construir implica planificação, preparação e execução
- Quanto mais complexo o sistema mais importante se torna a planificação e preparação



Leonor Melo

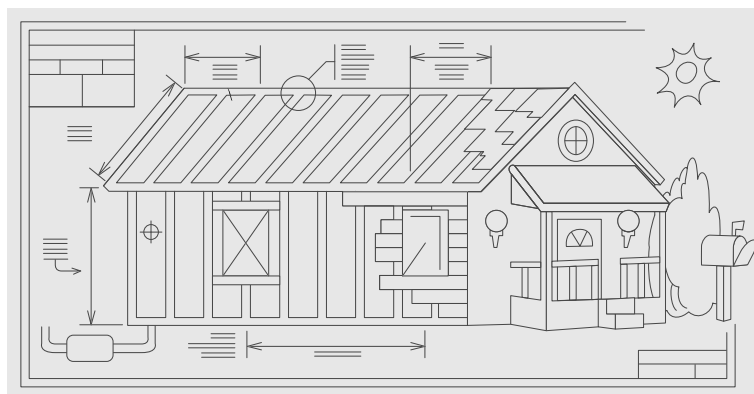
26 e 27 Processos de desenvolvimento de Software

5

5

## Construir software

- As consequências de um design infeliz num projeto mais complexo de construir são maiores



Leonor Melo

26 e 27 Processos de desenvolvimento de Software

6

6

## Construir software

- Construir edifício vs construir software:
  - Definir estilo de edifício
    - (definição problemas)
  - Plantas que apresentam várias perspetivas
    - (Design da arquitetura até design do software)
  - Feito por determinada ordem: fundações, paredes, instalações elétricas/água/gás, acabamentos
    - (várias fases do desenvolvimento e otimização)
  - Inspeções técnicas
    - (revisões do software)

Leonor Melo

26 e 27 Processos de desenvolvimento de Software

7

7

## Construir software

- Construir edifício vs construir software:
  - Nem tudo é construído de raiz: fogão, forno, exaustor, lavatórios, banheiras, etc. são comprados
    - (linguagens de alto-nível, bibliotecas, classes de interface com o utilizador e manipulação de BD pré-existent)
  - Certas partes podem ser adaptadas à medida: armários da cozinha, portas, janelas
    - (classes que sirvam de adaptadores de classes pré-existent para uma utilização mais uniforme)

Leonor Melo

26 e 27 Processos de desenvolvimento de Software

8

8

## Construir software

- Construir edifício vs construir software:
  - Nem todas as alterações têm o mesmo custo: mover certas paredes compromete a integridade do edifício
    - (nem todas as classes podem ser alteradas com a mesma facilidade)
- A importância da planificação, design e verificação variam de acordo com a dimensão e complexidade do projeto

Leonor Melo

26 e 27 Processos de desenvolvimento de Software

9

9

## Método waterfall

- Método de desenvolvimento de software waterfall (desenvolvimento em cascata)
  - Dividido por fases
  - Abordagem sequencial
    - Uma fase não pode começar antes da anterior terminar
  - Cada fase existe de forma isolada
    - O cliente deixa de ter qualquer papel no desenvolvimento uma vez terminada a fase inicial

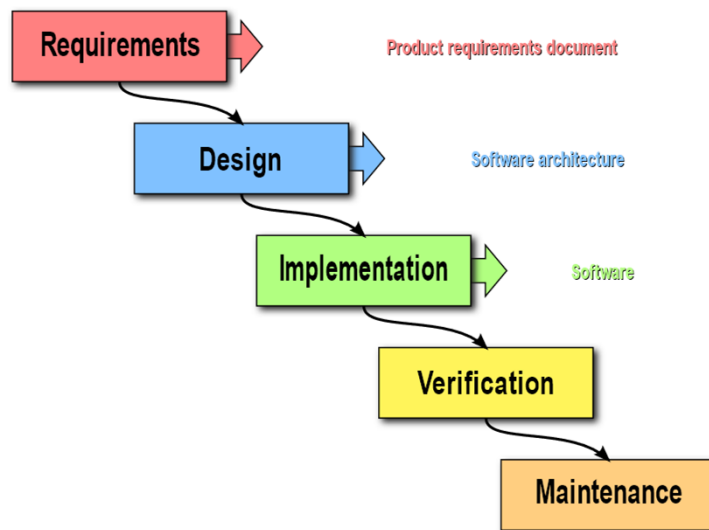
Leonor Melo

26 e 27 Processos de desenvolvimento de Software

10

10

## Método waterfall



Leonor Melo

26 e 27 Processos de desenvolvimento de Software

11

11

## Método waterfall

- Requisitos
  - Requisitos recolhidos logo no início do projeto
  - Restantes fases planeadas sem nenhuma intervenção adicional do cliente
- Design
  - Design lógico: brainstorming de possíveis soluções e teorias
  - Design físico: teorias e esquemas materializados em especificações concretas
- Implementação
  - Produção do código tendo por base os requisitos e especificações das fases anteriores

Leonor Melo

26 e 27 Processos de desenvolvimento de Software

12

12

## Método waterfall

- Verificação
  - Apresentação do produto terminado ao cliente.
- Manutenção
  - Utilização do produto por parte do cliente.
  - Cliente reporta bugs, características indesejadas e erros. A equipa de produção vai fazendo correções até ao cliente ficar satisfeito

Leonor Melo

26 e 27 Processos de desenvolvimento de Software

13

13

## Suposições do método waterfall

- Suposições do método waterfall:
  - Requisitos são conhecidos desde o início e com exatidão
  - Requisitos nunca (ou muito raramente) mudam
  - Os clientes sabem exatamente o que pretendem e não necessitam de ajuda a visualizar o sistema
  - O design pode ser feito de forma puramente abstrata e a tentativa raramente leva a erro

Leonor Melo

26 e 27 Processos de desenvolvimento de Software

14

14

## Suposições do método waterfall

- Suposições do método waterfall (cont.):
  - A tecnologia irá manter-se estável e responder como o esperado quando for feita a integração
  - O sistema não é assim tão complexo (e bonecos são para totós)
- Método waterfall:
  - Limita a habilidade de reagir a qualquer mudança ou de fazer correções ao longo do desenvolvimento

Leonor Melo

26 e 27 Processos de desenvolvimento de Software

15

15

## Motivo de falha de projetos de software

- Alguns dos motivos mais comuns que levam a falhas nos projetos de software são:
  - Falta de comunicação com o cliente
  - Requisitos mal definidos
  - Objetivos pouco realistas ou mal articulados
  - Falta de competência para lidar com a complexidade do projeto
  - Falta de organização
  - Práticas de desenvolvimento desleixadas
  - Estimativas pouco rigorosas

Leonor Melo

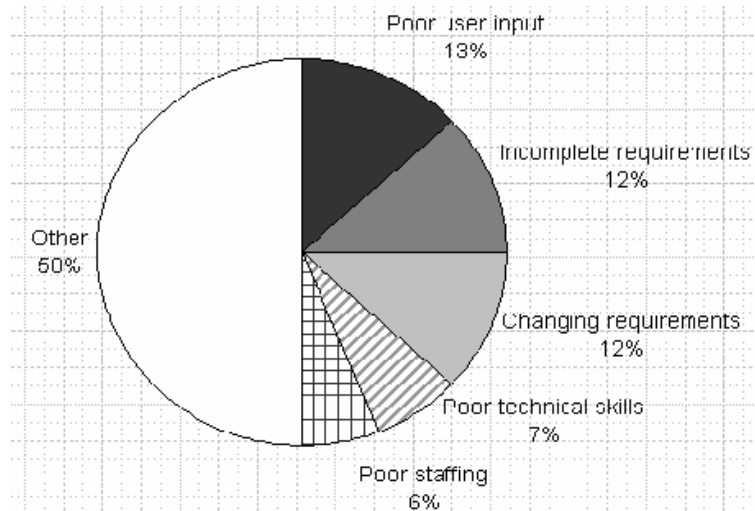
26 e 27 Processos de desenvolvimento de Software

16

16



## Motivo de falha de projetos de software



Leonor Melo

26 e 27 Processos de desenvolvimento de Software

17

17

## Motivo de falha de projetos de software

- (alguns) motivos de falha podem ser controlados:
  - Estimativas devem ser tão próximas da realidade quanto possível
  - Datas de entrega devem ser realistas
  - Riscos devem ser reconsiderados, controlados e geridos
  - Os trabalhadores podem ser recompensados por trabalhos extra

Leonor Melo

26 e 27 Processos de desenvolvimento de Software

18

18

## Necessidade de feedback e adaptação

- Sistemas complexos e sujeitos a mudança necessitam de feedback e capacidade de adaptação:
  - Feedback das implementações iniciais, da interpretação das especificações e das demonstrações ao cliente, para refinar os requisitos
  - Feedback dos testes e dos developers, para refinar o design e os modelos
  - Feedback do progresso da equipa desde as primeiras etapas, para refinar o planeamento e as estimativas
  - Feedback do cliente e do mercado, para re-priorizar as features a desenvolver na fase seguinte

Leonor Melo

26 e 27 Processos de desenvolvimento de Software

19

19

## Métodos ágeis

- Métodos ágeis
  - Não tem definição exata: práticas específicas variam bastante
- Normalmente:
  - aplicam desenvolvimento iterativo e incremental, de iterações curtas
  - usam planeamento adaptativo
  - promovem entregas incrementais
  - incluem práticas que promovem uma resposta rápida e flexível à mudança

Leonor Melo

26 e 27 Processos de desenvolvimento de Software

20

20

## Métodos ágeis

- Métodos ágeis (continuação)
  - Princípios e práticas que promovem:
    - Inclusão do cliente em todo o processo
    - Simplicidade
    - Leveza
    - Comunicação
    - Equipas auto-organizadas
  - Não está suposto de ser interpretado de forma extremista
    - Organização e reflexão continuam a ser necessárias

Leonor Melo

26 e 27 Processos de desenvolvimento de Software

21

21

## Unified Process

- Unified Software Development Process
- Processo de desenvolvimento de software
  - para criação e manutenção de sistemas orientados a objetos
  - iterativo
  - incremental
- Conjunto de boas práticas
  - Lifecycle iterativo
  - Guiado pelo risco
  - Descrição coesa e bem documentada

Leonor Melo

26 e 27 Processos de desenvolvimento de Software

22

22

## Unified Process

- Desenvolvimento iterativo e evolucionário
  - Permite começar o desenvolvimento com conhecimento incompleto e imperfeito
- Vantagens do desenvolvimento iterativo e evolucionário:
  - Progride de forma lógica para uma arquitetura robusta
  - Gestão eficaz de requisitos que vão mudando
  - Integração contínua
  - Compreensão precoce do sistema
  - Verificação do risco contínua

Leonor Melo

26 e 27 Processos de desenvolvimento de Software

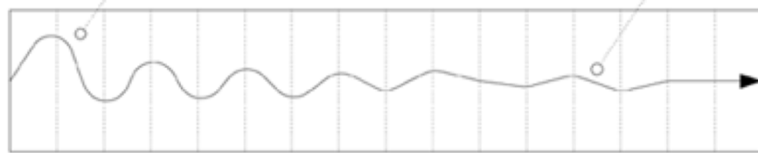
23

23

## Unified Process

Iterações iniciais com maiores desvios em relação aquilo que o sistema deve ser. Através do feedback e adaptação o sistema converge para os requisitos e design mais apropriados

Em iterações finais uma modificação significativa dos requisitos é rara mas não impossível. Alterações tardias podem dar uma vantagem competitiva à organização.



Uma iteração de design, implementação, integração e teste

Leonor Melo

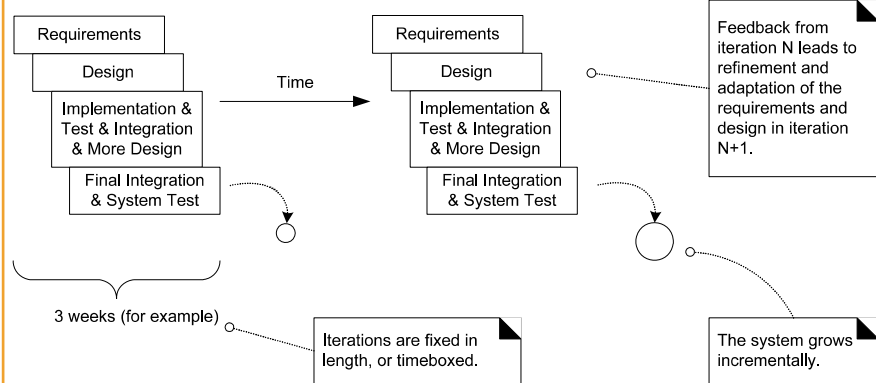
26 e 27 Processos de desenvolvimento de Software

24

24

## Unified Process

### • Desenvolvimento iterativo e evolucionário:



Leonor Melo

26 e 27 Processos de desenvolvimento de Software

25

25

## Unified Process

- O UP fornece
  - Orientação quanto à ordem de execução das tarefas da equipa
  - Integra o trabalho individual com o trabalho da equipa
  - Especifica os artefactos a produzir
  - Fornece critérios de medição e monitorização do progresso
- Quem faz o quê, quando e como?

Leonor Melo

26 e 27 Processos de desenvolvimento de Software

26

26

## RUP

- Rational Unified Process
  - Possivelmente versão mais conhecida do UP
    - Criada por uma divisão da IBM, que vende as ferramentas de apoio à metodologia
  - Promove a produtividade da equipa
  - Configurável
    - Nenhum processo de desenvolvimento é adequado a todos os sistemas de software
    - Adaptações para equipas grandes e pequenas
  - Documentação
    - Artefactos baseados em modelos
    - UML

Leonor Melo

26 e 27 Processos de desenvolvimento de Software

27

27

## RUP

- Peças fundamentais
  - Papéis (quem)
    - responsabilidades
  - Tarefas (como)
    - Unidades de trabalho
    - Orientado a resultados – tem de ser útil
  - Produtos do trabalho (o quê)
    - Produto resultante
  - Workflows (quando)
    - Quando é que as atividades devem ser executadas

Leonor Melo

26 e 27 Processos de desenvolvimento de Software

28

28

## Boas práticas do RUP

- Desenvolver o software de forma iterativa
  - Não é possível
    - Definir exaustivamente todo o problema antes de tudo o resto
    - Desenhar a solução completa
  - Cada iteração termina com uma release
- Gerir os requisitos
  - Casos de uso para capturar os requisitos funcionais
  - Devem ser claramente identificados e poder ser verificados

Leonor Melo

26 e 27 Processos de desenvolvimento de Software

29

29

## Boas práticas do RUP

- Incentiva arquiteturas baseadas em componentes
- Cria modelos visuais do software
  - Diversos tipo de modelo para comunicar
    - Diferentes aspetos do sistema, dependendo do leitor a que se destinam
  - UML
- Controlo das alterações feitas ao software
  - Integração contínua

Leonor Melo

26 e 27 Processos de desenvolvimento de Software

30

30

## Boas práticas do RUP

- Verificação da qualidade do software
  - Revisões
    - Requisitos funcionais
    - Requisitos não funcionais
  - Fazem parte do processo

Leonor Melo

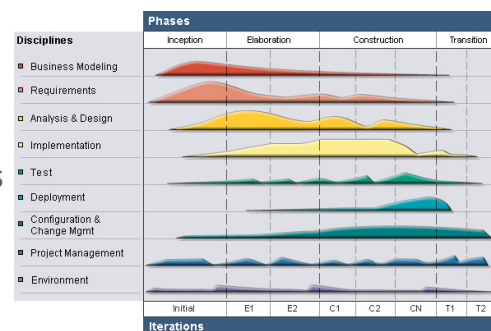
26 e 27 Processos de desenvolvimento de Software

31

31

## Processo RUP

- Eixo horizontal representa o tempo e mostra o aspeto dinâmico do processo
  - Expresso em termos de ciclos, fases, iterações e milestones
- Eixo vertical representa a vertente estática do processo
  - Descreve o processo em termos de atividades, artefactos, trabalhadores e workflows



Leonor Melo

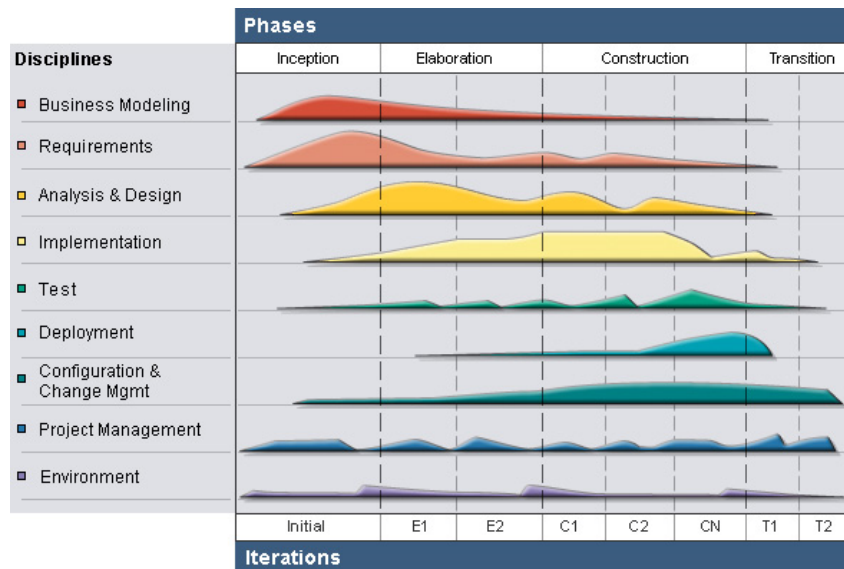
26 e 27 Processos de desenvolvimento de Software

32

32



## Processo RUP (exemplo para um pequeno projeto)



Leonor Melo

26 e 27 Processos de desenvolvimento de Software

33

33

## Fases e iterações

- Fases e iterações – o aspeto dinâmico do processo
  - O desenvolvimento é dividido em ciclos
    - Cada ciclo trabalha numa nova geração do produto
  - 4 fases:
    - *Inception* (ponto de início)
    - *Elaboração*
    - *Construção*
    - *Transição*

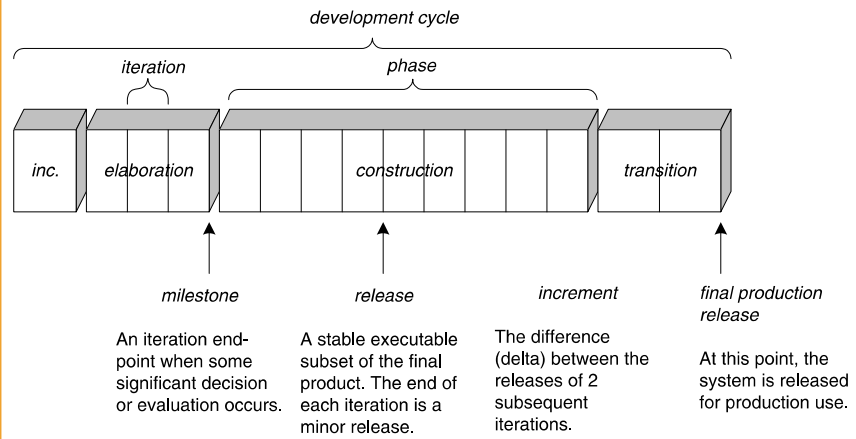
Leonor Melo

26 e 27 Processos de desenvolvimento de Software

34

34

## Fases e iterações



Leonor Melo

26 e 27 Processos de desenvolvimento de Software

35

35

## Fases do desenvolvimento

- Inception (início)
  - Âmbito do sistema
  - Intervenientes mais importantes
  - Risco, custo, etc.
- Elaboração
  - Identificação dos riscos
  - Domínio do problema
  - Análise e arquitetura
  - Continuação da identificação dos riscos

Leonor Melo

26 e 27 Processos de desenvolvimento de Software

36

36

## Fases do desenvolvimento

- Construção
  - Maior parte do software é construído nesta fase
  - Possivelmente dividido em subfases
- Transição
  - Transição de desenvolvimento para produção
  - Testes finais
  - Documentação

Leonor Melo

26 e 27 Processos de desenvolvimento de Software

37

37

## RUP principais workflows

- Modelação do negócio (business modeling)
  - O objetivo é compreender os conceitos e dinâmicas fundamentais do negócio desenvolvido pela organização (e que o sistema irá apoiar)
- Requisitos
  - O objetivo é definir o âmbito: o que será e o que não será desenvolvido
- Análise e Design
  - O objetivo é analisar os requisitos e desenhar uma solução

Leonor Melo

26 e 27 Processos de desenvolvimento de Software

38

38

## RUP principais workflows

- Implementação
  - O objetivo é criar o código baseado no design: desenvolvimento dos componentes
- Teste
  - O objetivo é garantir a qualidade através da verificação todos os aspetos do sistema
- Instalação (Deployment)
  - O objetivo é planear e entregar o sistema em funcionamento ao cliente

Leonor Melo

26 e 27 Processos de desenvolvimento de Software

39

39

## RUP workflows de suporte

- Gestão de projeto
- Gestão de configuração
- Envolvente (*environment*):
  - Selecionar e adquirir ferramentas
  - Construir ferramentas
  - Administração de Sistema
  - Treino

Leonor Melo

26 e 27 Processos de desenvolvimento de Software

40

40

## Exemplo de artefactos UP

Discipline	Artifact Iteration-*	Incep. Il	Elab. El. .En	Const. CL.Cn	Trans. T1..T2
Business Modeling	Domain Model		s		
Requirements	Use-Case Model	s	r		
	Vision	s	r		
	Supplementary Specification	s	r		
	Glossary	s	r		
Design	Design Model		s	r	
	SW Architecture Document		s		
	Data Model		s	r	
Implementation	Implementation Model		s	r	r
Project Management	SW Development Plan	s	r	r	r
Testing	Test Model		s	r	
Environment	Development Case	s	r		

Leonor Melo

26 e 27 Processos de desenvolvimento de Software

41

41

## inception

- Estabelecer o âmbito do software e as condições limites
  - Conceito operacional
  - Critério de aceitação
  - Descrição do que deve (e não deve) ser incluído
- Discriminar os casos de uso críticos do sistema
  - Comportamento dos cenários principais
- Propor (pelo menos) uma arquitetura candidata
- Estimar o custo global
- Estimar o risco

Leonor Melo

26 e 27 Processos de desenvolvimento de Software

42

42

## inception

- Atividades

- Formular o âmbito do projeto
- Planear e preparar o aspeto comercial e avaliar alternativas para gestão de risco, gestão de pessoal, planificação do projeto
- Sintetizar uma proposta de arquitetura

Leonor Melo

26 e 27 Processos de desenvolvimento de Software

43

43

## inception

- Resultados produzidos:

- Documento de visão
  - Visão geral dos requisitos fundamentais
  - Características principais
  - Restrições principais
- Todos os caso de uso, atores e *stakeholders* que podem ser identificados até ao momento
- Glossário do projeto inicial
- Proposta comercial incluindo
  - Contexto de negócio
  - Critério de sucesso
  - Previsão financeira
- Avaliação inicial de risco

Leonor Melo

26 e 27 Processos de desenvolvimento de Software

44

44

## inception

- Outros artefactos produzidos
  - Plano do projeto
    - Várias fases e iterações
  - Modelo de casos de uso inicial (10%-20% completo)
  - Modelo do domínio
  - Workflow do modelo de negócio (identificação dos processos de negócio, realizações, papéis e responsabilidades)
  - Descrição preliminar do processo de desenvolvimento
  - Um ou mais protótipos
    - Comportamento, estrutura, ...

Leonor Melo

26 e 27 Processos de desenvolvimento de Software

45

45

## inception

- Milestone da inception:
  - Concordância dos stakeholders quanto à definição do âmbito e estimativa inicial de custo e calendário (que serão refinados em fases posteriores)
  - Concordância quanto a que o conjunto certo de requisitos foi capturado e que existe uma conceptualização partilhada sobre esses requisitos
  - Concordância quanto à adequação das estimativas de custo e calendário, prioridades, risco e processo de desenvolvimento
  - Concordância quanto à identificação dos riscos iniciais e à existência de uma estratégia de mitigação para cada um deles

Leonor Melo

26 e 27 Processos de desenvolvimento de Software

46

46

## Elaboração

- Objetivos

- Analisar o domínio do problema
- Desenvolver o plano do projeto
- Consolidar e detalhar os casos de uso mais críticos (os que terão maior impacto para que o projeto seja considerado um sucesso)
  - Mitigar os elementos de risco elevado e produzir um calendário e estimativa de custo mais rigoroso

Leonor Melo

26 e 27 Processos de desenvolvimento de Software

47

47

## Elaboração

- Objetivos

- Elaborar um plano para a fase de construção
- Estabelecer uma fundação sólida para a arquitetura
  - Definir, validar e chegar a consenso sobre a arquitetura tão cedo quanto possível
  - Demonstrar que a arquitetura irá suportar o sistema delineado no documento de visão, mantendo um custo e um gasto de tempo razoáveis

Leonor Melo

26 e 27 Processos de desenvolvimento de Software

48

48



## Elaboração

- Atividades
  - Desenhar arquitetura e selecionar componentes de forma iterativa:
    - Avaliar potenciais componentes
    - Integrar os componentes e re-avaliar
  - Decisões de fazer / comprar / reutilizar vão afetar o custo e duração da fase de construção

Leonor Melo

26 e 27 Processos de desenvolvimento de Software

49

49

## Elaboração

- Resultados produzidos
  - Modelo de caso de uso (80% completo)
    - Todos os casos de uso identificados
    - Todos os atores identificados
    - A maior parte das descrições de caso de uso escritas
  - Requisitos suplementares
    - Não funcionais ou não associados com um caso de uso
  - Descrição da arquitetura do software

Leonor Melo

26 e 27 Processos de desenvolvimento de Software

50

50

## Elaboração

- Resultados produzidos
  - Protótipo de arquitetura executável
  - Lista de riscos e caso de negócio revistos
  - Plano de desenvolvimento para todo o projeto
    - Projeto de plano genérico com as iterações e critérios de avaliação para cada iteração
  - Especificação do processo de desenvolvimento atualizado
  - Manual do utilizador preliminar (opcional)

Leonor Melo

26 e 27 Processos de desenvolvimento de Software

51

51

## Elaboração

- Milestones da elaboração
  - A visão e os requisitos estão estáveis?
  - A arquitetura está estável?
  - As abordagens a usar para teste e avaliação estão confirmadas?
  - Os testes e avaliação dos protótipos executáveis demonstraram que os elementos de maior risco foram considerados e resolvidos?
  - O plano de iterações para a fase de construção é suficientemente detalhado e realista para que o trabalho possa prosseguir?

Leonor Melo

26 e 27 Processos de desenvolvimento de Software

52

52

## Elaboração

- Milestones da elaboração

- Todos os stakeholders concordam que a visão documentada no documento de visão pode ser alcançada se o plano atual for executado de forma a desenvolver o sistema de acordo com a arquitetura atual?
- Os custo atuais são aceitáveis face aos custos planeados?

## Construção

- Todos os componentes e características da aplicação restantes são desenvolvidas e integradas no produto
- Todas as características são testadas de forma minuciosa
- O ênfase é na gestão dos recurso e controlo das operações de forma a otimizar o custo, calendário e qualidade
- Construção paralela pode acelerar a disponibilização de releases

## Construção

- Objetivos:
  - Minimizar o custo de desenvolvimento
    - Otimizar recursos
    - Evitar deita-fora-e-recomeça desnecessário
  - Alcançar qualidade adequada tão rapidamente quanto possível
  - Alcançar versões úteis (alfa, beta, ou outras) tão rapidamente quanto possível

## Construção

- Atividades:
  - Gestão de recursos, controlo de recursos, otimização do processo
  - Desenvolvimento de componentes completos e testados de acordo com os critérios de avaliação definidos antes
  - Verificação do produto lançado em relação aos critérios de aceitação documentados na visão

## Construção

- Resultados produzidos:
  - Um produto pronto a ser entregue ao utilizador final
  - Um produto integrado nas plataforma adequadas
  - Os manuais do utilizador
  - A descrição da versão atual

Leonor Melo

26 e 27 Processos de desenvolvimento de Software

57

57

## Construção

- Milestones da construção
  - Esta versão do produto está estável e madura o suficiente para ser lançada na comunidade de utilizadores?
  - Estão todos os stakeholders prontos para a transição para a comunidade de utilizadores?
  - Os custos atuais quando comparados com os custos planeados continuam aceitáveis?
- A transição pode ter de ser adiada uma versão se alguma destas perguntas não for respondida afirmativamente

Leonor Melo

26 e 27 Processos de desenvolvimento de Software

58

58

## Transição

- Disponibiliza o projeto de software à comunidade de utilizadores
- Depois do lançamento, normalmente aparecem problemas que requerem novas versões ou o terminar de características do software que foram sendo adiadas
- Esta fase começa quando a versão base está madura o suficiente para ser lançada no domínio do utilizador final
  - Uma versão utilizável e razoavelmente completa foi construída com um nível aceitável de qualidade e existe documentação disponível

Leonor Melo

26 e 27 Processos de desenvolvimento de Software

59

59

## Transição

- Objetivos
  - Garantir que os utilizadores conseguem usar o sistema de forma autónoma
  - Obter a concordância do stakeholders de que o deployment do sistema está completo e de acordo com os critérios de avaliação da visão
  - Alcançar o produto final base de forma tão rápida e económica possível

Leonor Melo

26 e 27 Processos de desenvolvimento de Software

60

60

## Transição

- Faz parte da fase de transição
  - Teste à versão beta para validar o sistema em relação às expectativas dos utilizadores
  - Funcionamento em paralelo com o legacy system que o projeto irá substituir
  - Conversão de base de dados operacionais
  - Treino de utilizadores e responsáveis pela manutenção
  - Envio do produto para as equipas de marketing, distribuição e vendas
- Termina quando deployment completa a visão

Leonor Melo

26 e 27 Processos de desenvolvimento de Software

61

61

## Transição

- Objetivos
  - Garantir que os utilizadores se conseguem usar o sistema de forma autónoma
  - Obter a concordância do stakeholders de que o deployment do sistema está completo e de acordo com os critérios de avaliação da visão
  - Alcançar o produto final base de forma tão rápida e económica possível

Leonor Melo

26 e 27 Processos de desenvolvimento de Software

62

62

## Transição

- Atividades
  - Ações específicas do deployment em questão: produção e empacotamento comercial, lançamento de vendas, treino do pessoal
  - Afinação de detalhes, incluindo correção de erros e melhorias de performance e usabilidade
  - Verificação do deployment em relação à visão e critérios de aceitação do produto

Leonor Melo

26 e 27 Processos de desenvolvimento de Software

63

63

## Transição

- Atividades (cont.)
  - As atividades dependem do objetivo:
    - Para corrigir bugs
      - Implementação e teste são geralmente suficientes
    - Para novas features
      - A iteração é semelhante a uma da fase de construção

Leonor Melo

26 e 27 Processos de desenvolvimento de Software

64

64



## Transição

- Milestones da transição:
  - Os clientes estão satisfeitos?
  - Os custos atuais quando comparados com os planeados são aceitáveis?
    - Se não, que ações executar em projetos futuros para evitar esse problema?