Laboratório de Engenharia de Software

Especificações

Arndt von Staa
Departamento de Informática
PUC-Rio
Março 2015

Laboratório de Engenharia de Software

Especificação

- Objetivo da aula
  - esboçar algumas técnicas de especificação, dando ênfase a casos de uso
    - assume-se que os alunos já tenham visto, ou irão ver, isto em mais detalhe em alguma outra disciplina
- Justificativa
  - através da adoção de alguns padrões de redação, pode-se reduzir significativamente os defeitos em especificações
  - casos de uso podem dirigir o desenvolvimento dos testes
    - desenvolvimento dirigido por comportamento (BDD - *behavior driven development*)
  - casos de uso podem ser aplicados a
    - interação do usuário (pessoa) com o artefato
    - interação entre artefatos

Leitura complementar: [Cockburn, 2005]

Especificação textual convencional

- Uma grande parte das especificações é redigida sob a forma de textos sem uma estrutura padronizada
- Quais seriam os problemas com esta abordagem?
  - São uma das principais causas da entrega de sistemas inadequados aos usuários
    - há autores que dizem que 70% dos defeitos de software entregue são consequência de erros nas especificações
  - São frequentemente ambíguas, incompletas e incoerentes
  - Dificultam o controle da qualidade
  - Não deixam claro:
    - qual a totalidade do serviço a ser prestado pelo software
    - tampouco informam os requisitos não funcionais
    - vários (potenciais) interessados não são consultados

sistema inadequado – é um sistema que não satisfaz os interesses ou necessidades do usuário

Historieta (*extreme programming*)

- Historieta (*user story*) – pequena estória, algo que caiba em uma ficha catalográfica
  - historieta é sempre uma ação
  - historieta sempre tem um nome: o objetivo principal
- Estrutura da frase de uma historieta (simples):
  - quem?
  - faz o que?
  - por que faz?
- Como ficaria a historieta "Registrar esforço" ?
- Como desenvolvedor quero registrar em *folhas de tempo* o tempo gasto por tarefa realizada em determinado dia para saber onde despendi o meu esforço naquele dia.

Historieta

Laboratório de Engenharia de Software

Historietas podem utilizar elementos que necessitam de uma especificação mais detalhada. Ex.

- o que é uma folha de tempo?

O texto a seguir precisa de explicação (decomposição)?

- Cada folha de tempo registra o tempo gasto (esforço) por tarefa e atividade realizada em um determinado dia
  - Definição: uma atividade leva a um artefato de qualidade controlada.
  - Definição: uma tarefa é um passo de trabalho na direção de concluir uma atividade.
  - Exemplo:
    - atividade: *desenvolver um módulo*
    - tarefas: *projetar o módulo*, *redigir o módulo de definição*, *redigir o módulo específico de teste*, *redigir o script de teste inicial*, *redigir o módulo de instrumentação*, *redigir o código do módulo*, *testar o módulo*, *redigir o script de teste final*, *assegurar total conformidade com os padrões requeridos*, *aceitar o módulo*

Historieta

Laboratório de Engenharia de Software

- Uma atividade leva a um artefato de qualidade controlada.
- Uma tarefa é um passo de trabalho na direção de concluir uma atividade.
  - Podem ser usadas para medir? O que?
    - produção
    - esforço
    - produtividade
  - O que falta para poder medir corretamente?
    - Regras de negócio
      - atividades são realizadas por uma ou mais tarefas.
      - as tarefas de uma mesma atividade podem ocorrer em diferentes dias.
      - uma mesma tarefa pode ocorrer repetidas vezes em um mesmo dia ou em dias diferentes.
      - uma atividade estará concluída somente quando o artefato que visa desenvolver tiver sido aceito.

Crítica à estrutura da frase

- A estrutura da frase usada até agora é sempre suficiente?
  - Como especificar coisas repetidas?
    - ex. para cada tarefa devem ser fornecidos ...
  - Como especificar inicializações?
    - ex. ao entrar no sistema o desenvolvedor quer ver a folha de tempo do dia corrente com todos os dados preenchidos até o momento, inclusive os pré-preenchidos
  - Mais alguma coisa?
    - ex. a lista de atividades pendentes, i.e. as que já deveriam estar concluídas e as que deveriam terminar no dia mais próximo do dia corrente
      - deu para entender isso?
  - Requisitos não funcionais?

Historieta mais elaborada

Estilo mais elaborado:

- [<contexto: condições>]<ator: quem, ou o que> <ação: quer, ou faz><objetos: o que – objetos diretos e indiretos conforme requeridos pelo verbo><razão: por que>[<condições>]

[x] - x é opcional

Exemplo de historietas mais elaboradas

- Exemplo 1
- Uma forma "impessoal de redigir"
  - O desenvolvedor deseja saber o esforço gasto por atividade para que possa aprimorar o seu modo de trabalhar.
- Uma forma de redigir focada na pessoa
  - Como desenvolvedor desejo saber o esforço gasto pelas tarefas de uma atividade para que eu possa aprimorar o meu modo de trabalhar.

Exemplo de historietas mais elaboradas

- Exemplo 2
  - Como desenvolvedor, de modo que possa determinar como gasto meu esforço, informarei para cada tarefa efetuada, a hora inicial, a hora final ou a duração, a natureza da tarefa, a atividade a que pertence, e observações relativas à tarefa.
  - Caso eu informe a hora inicial e a final, a duração será calculada. Caso informe a hora inicial e a duração, a hora final será calculada.

Exemplos de historietas mais elaboradas

- Para poupar tempo ao preencher,
  - o desenvolvedor quer ver a folha de tempo do dia corrente sempre que ativar o sistema.
- Para organizar o seu trabalho diário,
  - o desenvolvedor deseja ver os dados relativos às atividades pendentes registradas na folha correspondente ao dia selecionado.
- Para poder relembrar ou mesmo corrigir dados incorretos,
  - o desenvolvedor deseja poder selecionar para ler ou editar qualquer uma das folhas de tempo de dias passados até o dia corrente inclusive.

Avaliação

- Com uma especificação completa redigida neste estilo você sabe o que deverá ser desenvolvido?
  - você sabe quais são as finalidades (serviço) do programa?
  - você consegue identificar as características (*features*) que constituem o programa?
    - cada historieta corresponde a exatamente uma característica?
  - você sabe quais são as interações com outros sistemas já previstas?
  - você sabe quais os requisitos não funcionais a serem satisfeitos?
  - você sabe como interagir com o sistema?
  - você sabe, ou é capaz de imaginar, que erros de uso podem ocorrer?
    - você sabe como erros de uso devem ser tratados?

Algumas recomendações

- Historietas (*user stories*) são muitas vezes incompletas
  - o desenvolvedor não deve completar da forma que acha, mas sim, deve procurar o autor (*product owner* na terminologia SCRUM) para tirar dúvidas
    - problema, o autor estará sempre presente?
    - se estiver presente um representante (*proxy*) este tem suficiente conhecimento para poder responder?
- Historietas interdependem
  - o tratamento isolado e, possivelmente, em tempos não coincidentes consegue assegurar que a solução dada para uma não conflita com a solução dada para outra?

Stellman, A.; Greene, J.; *2014. Learning Agile: Understanding Scrum, XP, Lean, and Kanban.* O'Reilly Media. Kindle Edition.

- É uma excelente referência para quem quer aprender processos ágeis

Componente *Controle de Acesso* 1 / 3

- O *Controle de Acesso* é um **componente genérico** a ser utilizado por um sistema qualquer, digamos **`Sis`**, para identificar usuários que, ao realizarem o *login* no sistema **`Sis`**, são autorizados com os seus direitos de uso
- Um usuário autorizado identifica-se por meio de um **`idUsuario`** e de uma **`Senha`** (qual seria a forma de redação padrão?)
- Cada par **`<idUsuario, Senha>`** vincula-se a exatamente um conjunto de direitos de uso
  - O conjunto de todos os possíveis direitos de uso é específico para cada sistema **`Sis`** que venha a utilizar o componente *Controle de Acesso*
- Um mesmo usuário pode possuir diversas senhas em um mesmo sistema, cada uma associada a um determinado conjunto de direitos de uso

`Sis` é uma variável que receberá o nome do sistema específico

Ex: Componente *Controle de Acesso* 2 / 3

- O cadastro de usuários autorizados é mantido pelo gerente do *Controle de Acesso* do sistema **`Sis`**.
- A cada par **`<idUsuario, Senha>`** no cadastro o gerente associa o conjunto de direitos de uso.
  - o usuário não pode alterar os direitos de uso
  - o usuário pode alterar a senha, porém não pode utilizar a mesma senha para diferentes conjuntos de direitos de uso
  - caso o usuário tenha esquecido a senha, poderá solicitar o envio de nova senha por mensagem eletrônica para o endereço eletrônico cadastrado inicialmente pelo gerente do *Controle de Acesso*

Ex: Componente *Controle de Acesso* 3 / 3

Esboço da interface gráfica do componente *Controle de Acesso*

- Os caracteres de controle *Captcha* têm por objetivo reduzir a chance de um robô tentar e conseguir interagir com o sistema
- A cada vez que o usuário puder entrar com dados,
  - os campos de dados deverão estar vazios
  - deverá ser fornecido um novo conjunto de caracteres de controle escolhidos randomicamente

Sistema **`Sis`**

Usuário

Senha

Digite os caracteres Captcha

Login | Cancelar | Mudar senha | Esqueci senha

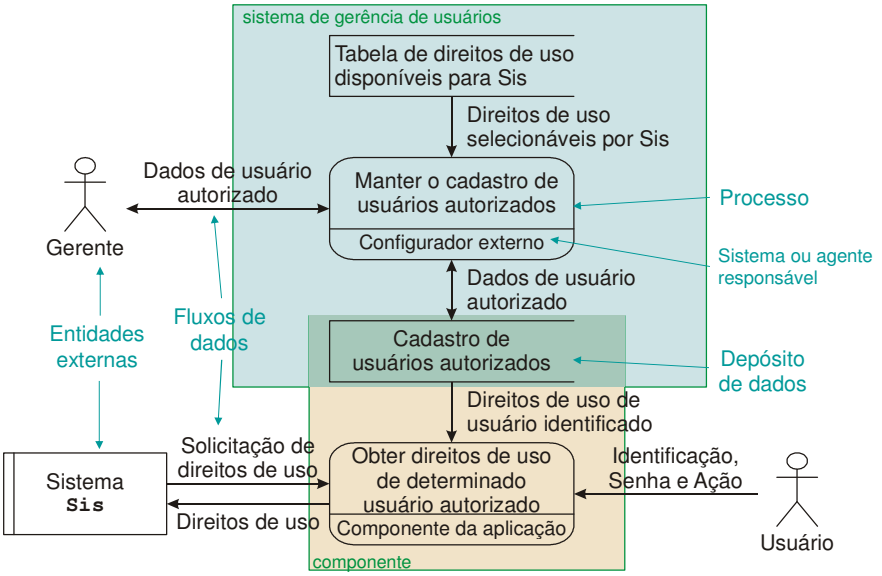
Obs. Para diminuir ações erradas por parte do usuário, os nomes do *widgets* devem dar uma boa ideia do seu propósito.

Diagramas de fluxo de dados

- Diagramas de fluxo de dados (DFD) podem especificar a arquitetura identificando a interdependência entre
  - processos
  - entidades externas
    - e.g. pessoas, outros sistemas, outros componentes
  - depósitos de dados
    - e.g. arquivos, bases de dados, tabelas
    - comunicação assíncrona
  - dados que fluem entre os elementos acima
    - comunicação síncrona
- Permitem particionar um sistema complexo em partes (processos) estanques com interfaces bem definidas
  - possível instrumento para projetar boas arquiteturas

Ao invés de DFDs poderiam ser utilizados diagramas de atividades da UML 2

Arquitetura do sistema *Controle de Acesso*



Qualidade das especificações

- Do ponto de vista da qualidade precisamos:
  - *poder controlar a qualidade das especificações antes de iniciar o desenvolvimento*
    - com vistas a reduzir a quantidade de defeitos possivelmente contidos nas especificações
    - evitar defeitos decorrentes da falha de entendimento do que o usuário precisa ou deseja
      - lembre-se: solução correta do problema errado não interessa ao usuário

Laboratório de Engenharia de Software

Qualidade das especificações

- Do ponto de vista da qualidade precisamos:
  - *poder controlar após desenvolver de modo a determinar se o que foi implementado corresponde ao especificado*
  - poder criar suítes de teste a partir da especificação antes de desenvolver
    - assegura a verificabilidade da especificação
    - contribui para o desenvolvimento dirigido por testes
    - a suíte de teste existe antes de desenvolver e passa a ser uma forma de especificar baseada em exemplos
    - a suíte de teste visando teste automatizado viabiliza o controle a baixo custo durante o desenvolvimento incremental

Laboratório de Engenharia de Software

Casos de uso: o que são

- Casos de uso são uma das possíveis formas de especificar requisitos
  - fazem parte do conjunto de linguagens de representação da UML – *Unified Modeling Language*
  - são utilizados com bastante frequência
  - tendem a ser excessivamente informais
    - adicionamos mais estrutura e formalidade para reduzir as deficiências da proposta original

Cockburn, A.; Escrevendo Casos de Uso Eficazes - Um Guia para Desenvolvedores de Software; São Paulo, SP: Bookman; 2005

Casos de uso: o que são

- Casos de uso viabilizam o desenvolvimento dirigido por características (*feature driven development*)
  - uma característica (*feature*) é uma funcionalidade única bem delimitada visando um determinado propósito
    - funcionalidade pode ser definida em níveis de abstração
    - pode ser estendido para corresponder a requisitos não funcionais
  - cada característica deve corresponder a exatamente um propósito
  - características podem existir em vários níveis de abstração
  - para o nosso caso desejamos características de baixo nível de abstração

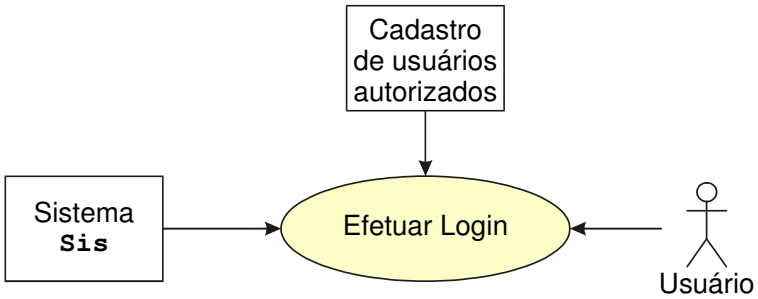
Casos de uso: como criá-los

- Processo coerente com processos ágeis de desenvolvimento, ex: Scrum e *eXtreme Programming*
  - identificam-se as características (de baixo nível de abstração), possivelmente a partir de historietas
  - ordenam-se as características segundo a relevância para o usuário e, se necessário, observando as precedências
  - da característica mais relevante para a menos relevante
    - desenvolve-se incrementalmente o sistema adicionando características aceitas
    - ao término do desenvolvimento de uma característica, o usuário avalia o sistema parcialmente desenvolvido (construto, *build*) e verifica se atende aos seus anseios atuais
  - possivelmente adicionam-se novas e/ou eliminam-se características definidas e ainda não implementadas
  - reavaliam-se a relevância das características ainda a implementar e repete a partir de 2 até concluir a liberação

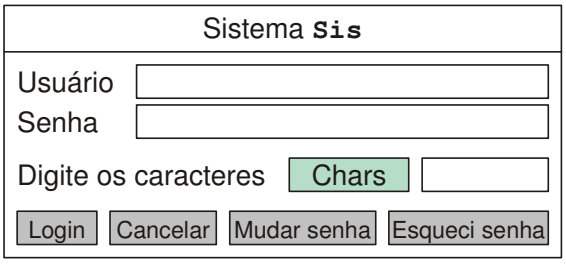
Caso de uso por característica

- Cada caso de uso deve descrever exatamente uma característica (propósito)
  - exatamente: nem mais, nem menos *Goldilocks principle*: just right
- A implementação de um caso de uso pode envolver diversos programas, por exemplo cliente (browser) e servidor
  - Exemplo
    - ao fornecer os dados para um sistema
      - evidentemente não estamos interessados somente nos dados digitados pelo cliente
      - estamos interessados também na validação desses dados
      - mas a validação possivelmente requer acesso a bases de dados, o que deverá ser realizado no servidor

Componente Controle de Acesso: Login



Esboço ➔



Crítica aos diagramas de caso de uso

- O que significa o diagrama?
  - quais são os dados?
  - quais são os resultados?
  - qual a finalidade ou propósito?
- Como verificar se a implementação está correta?
  - quais são os casos de teste?
  - quais são os critérios de aceitação?
  - quais são as regras de negócio?
- Como gerar os casos de teste?
- Qual o conjunto de todos os casos de uso do componente ou sistema?

Crítica aos diagramas de caso de uso

- Evidentemente a linguagem gráfica proposta é excessivamente abstrata
  - solução: utilizar um formulário para especificar os detalhes de cada caso de uso
    - o formulário a seguir foi inspirado em [Cockburn, 2001]

Exemplo: Caso de uso Efetuar Login

Caso de uso	
Resumo	
Escopo	
Ator principal	
Interessados	
Invariante	
Pré condição	
Acionamento	

Exemplo: Caso de uso Efetuar Login

Laboratório de Engenharia de Software

Caso de uso	Efetuar login no sistema `Sis`
Resumo	
Escopo	
Ator principal	
Interessados	
Invariante	
Pré condição	
Acionamento	

Exemplo: Caso de uso Efetuar Login

Laboratório de Engenharia de Software

Caso de uso	Efetuar login no sistema `Sis`
Resumo	O sistema `Sis` solicita que o componente Controle de Acesso forneça os *direitos de uso* correspondente ao par identificação e senha previamente cadastrado digitado pelo usuário
Escopo	
Ator principal	
Interessados	
Invariante	
Pré condição	
Acionamento	

Exemplo: Caso de uso Efetuar Login

Caso de uso	Efetuar login no sistema **`Sis`**
Resumo	O sistema **`Sis`** solicita que o componente Controle de Acesso forneça os *direitos de uso* correspondente ao par identificação e senha previamente cadastrado digitado pelo usuáriouar segundo o correspondente conjunto de *direitos de uso*
Escopo	Componente que visa obter autorização de uso
Ator principal	
Interessados	
Invariante	
Pré condição	
Acionamento	

Exemplo: Caso de uso Efetuar Login

Caso de uso	Efetuar login no sistema **`Sis`**
Resumo	O sistema **`Sis`** solicita que o componente Controle de Acesso forneça os *direitos de uso* correspondente ao par identificação e senha previamente cadastrado digitado pelo usuário
Escopo	Componente que visa obter autorização de uso
Ator principal	Sistema **`Sis`**
Interessados	
Invariante	
Pré condição	
Acionamento	

Exemplo: Caso de uso Efetuar Login

Caso de uso	Efetuar login no sistema **`Sis`**
Resumo	O sistema **`Sis`** solicita que o componente Controle de Acesso forneça os *direitos de uso* correspondente ao par identificação e senha previamente cadastrado digitado pelo usuário
Escopo	Componente que visa obter autorização de uso
Ator principal	Sistema **`Sis`**
Interessados	Sistema **`Sis`** Objetivo: permitir somente a usuários autorizados o uso segundo um determinado conjunto de direitos de uso Usuário Objetivo: obter a autorização de uso segundo os direitos de uso com os quais foi registrado Cadastro de usuários autorizados Objetivo: registrar cada par <usuário, senha> autorizado e respectivos direitos de uso
Invariante	
Pré condição	
Acionamento	

Laboratório de Engenharia de Software

Exemplo: Caso de uso Efetuar Login

Caso de uso	Efetuar login no sistema **`Sis`**
Resumo	O sistema **`Sis`** solicita que o componente Controle de Acesso forneça os *direitos de uso* correspondente ao par identificação e senha previamente cadastrado digitado pelo usuário
Escopo	Componente que visa obter autorização de uso
Ator principal	Sistema **`Sis`**
Interessados	Sistema **`Sis`** Objetivo: permitir somente a usuários autorizados o uso segundo um determinado conjunto de direitos de uso Usuário Objetivo: obter a autorização de uso segundo os direitos de uso com os quais foi registrado Cadastro de usuários autorizados Objetivo: registrar cada par <usuário, senha> autorizado e respectivos direitos de uso
Invariante	O cadastro de usuários autorizados está atualizado, disponível e criptografado segundo uma chave interna
Pré condição	
Acionamento	

Laboratório de Engenharia de Software

Exemplo: Caso de uso Efetuar Login

Laboratório de Engenharia de Software

Caso de uso	Efetuar login no sistema **Sis**
Resumo	O sistema **Sis** solicita que o componente Controle de Acesso forneça os *direitos de uso* correspondente ao par identificação e senha previamente cadastrado digitado pelo usuário
Escopo	Componente que visa obter autorização de uso
Ator principal	Sistema **Sis**
Interessados	Sistema **Sis** Objetivo: permitir somente a usuários autorizados o uso segundo um determinado conjunto de direitos de uso Usuário Objetivo: obter a autorização de uso segundo os direitos de uso com os quais foi registrado Cadastro de usuários autorizados Objetivo: registrar cada par <usuário, senha> autorizado e respectivos direitos de uso
Invariante	O cadastro de usuários autorizados está atualizado, disponível e criptografado segundo uma chave interna
Pré condição	- - -
Acionamento	

Exemplo: Caso de uso Efetuar Login

Laboratório de Engenharia de Software

Caso de uso	Efetuar login no sistema **Sis**
Resumo	O sistema **Sis** solicita que o componente Controle de Acesso forneça os *direitos de uso* correspondente ao par identificação e senha previamente cadastrado digitado pelo usuário
Escopo	Componente que visa obter autorização de uso
Ator principal	Sistema **Sis**
Interessados	Sistema **Sis** Objetivo: permitir somente a usuários autorizados o uso segundo um determinado conjunto de direitos de uso Usuário Objetivo: obter a autorização de uso segundo os direitos de uso com os quais foi registrado Cadastro de usuários autorizados Objetivo: registrar cada par <usuário, senha> autorizado e respectivos direitos de uso
Invariante	O cadastro de usuários autorizados está atualizado, disponível e criptografado segundo uma chave interna
Pré condição	- - -
Acionamento	*Efetuar login* inicia quando o sistema **Sis** solicitar pela primeira vez o fornecimento de uma autorização de uso

Exemplo: Caso de uso Efetuar Login

Fluxo principal	
Fluxos alternativos	

Exemplo: Caso de uso Efetuar Login

Fluxo principal	1. O componente limpa os campos e gera os caracteres de controle 2. O usuário digita sua identificação, senha e os caracteres de controle 3. O usuário clica "Login" 4. O controle de acesso verifica se os dados estão lexicamente corretos 5. O controle de acesso verifica se <usuário, senha> está autorizado 6. O controle de acesso retorna ao sistema **Sis**, fornecendo a condição "autorizar uso" e os direitos de uso correspondentes a <usuário, senha>
Fluxos alternativos	

Exemplo: Caso de uso Efetuar Login

Fluxo principal	1. O componente limpa os campos e gera os caracteres de controle 2. O usuário digita sua identificação, senha e os caracteres de controle 3. O usuário clica "OK" 4. O controle de acesso verifica se os dados estão lexicamente corretos 5. O controle de acesso verifica se <usuário, senha> está autorizado 6. O controle de acesso retorna ao sistema **Sis**, fornecendo a condição "autorizar uso" e os direitos de uso correspondentes a <usuário, senha>
Fluxos alternativos	Evento E1/4: Um ou mais campos estão lexicamente incorretos E1.1 Até 3 tentativas efetue E1.1.1 O controle de acesso informa ao usuário a ocorrência de um erro de uso E1.1.2 repete a partir de 1. E1.2 Após a terceira tentativa efetue E1.2.1 O controle de acesso informa ao usuário que ele não está autorizado a usar o sistema Sis E1.2.2 O controle de acesso fornece a condição "não autorizado" e os direitos "vazio" ao sistema Sis Fim repetição Fim evento E1.

Exemplo: Caso de uso Efetuar Login

Fluxos alternativos	

Exemplo: Caso de uso Efetuar Login

Laboratório de Engenharia de Software

Fluxos alternativos	Evento E2/5: o usuário digita um par <usuário, senha> que não é confirmado E2.1 Até 3 tentativas efetue E2.1.1 O controle de acesso informa ao usuário a ocorrência de um erro de uso E2.1.2 Repete a partir de 1. E2.2 Após a terceira tentativa efetue E2.2.1 O controle de acesso informa ao usuário que ele não está autorizado a usar o sistema **Sis** E2.2.2 O controle de acesso fornece o a condição "não autorizado" e os direitos "vazio" ao sistema Sis Fim repetição Fim evento E2.

Exemplo: Caso de uso Efetuar Login

Laboratório de Engenharia de Software

Fluxos alternativos	Evento E2/5: o usuário digita um par <usuário, senha> que não é confirmado E2.1 Até 3 tentativas efetue E2.1.1 O controle de acesso informa ao usuário a ocorrência de um erro de uso E2.1.2 Repete a partir de 1. E2.2 Após a terceira tentativa efetue E2.2.1 O controle de acesso informa ao usuário que ele não está autorizado a usar o sistema **Sis** E2.2.2 O controle de acesso fornece o conjunto "não autorizado" ao sistema **Sis** Fim repetição Fim evento E2.
	Evento E3/3: o usuário clica "Trocar senha" E3.1 O controle de acesso verifica se os dados estão lexicamente corretos E3.2 O controle de acesso verifica se <usuário, senha> está autorizado E3.3 O controle de acesso ativa o diálogo "Trocar senha" E3.4 Repete a partir de 1 Fim evento E3.

Exemplo: Caso de uso Efetuar Login

Laboratório de Engenharia de Software

Fluxos alternativos	

Exemplo: Caso de uso Efetuar Login

Laboratório de Engenharia de Software

Fluxos alternativos	Evento E4/3: o usuário clica "Esqueci senha" E4.1 O controle de acesso verifica se os dados estão lexicamente corretos E4.2 O controle de acesso verifica se usuário está cadastrado E4.3 O controle de acesso ativa o diálogo "Esqueci senha" E4.4 Se retornar do diálogo E4.4.1 Repete a partir de 1 Fim se Fim evento E4.

Exemplo: Caso de uso Efetuar Login

Fluxos alternativos	Evento E4/3: o usuário clica "Esqueci senha" E4.1 O controle de acesso verifica se os dados estão lexicamente corretos E4.2 O controle de acesso verifica se usuário está cadastrado E4.3 O controle de acesso ativa o diálogo "Esqueci senha" E4.4 Se retornar do diálogo E4.4.1 Repete a partir de 1 Fim se Fim evento E4.
	Evento E5: o usuário clica "Cancelar" em qualquer lugar E5.1 O sistema solicita confirmação do cancelamento E5.2 Se usuário confirma o cancelamento E5.2.1 O controle de acesso retorna fornecendo o conjunto "cancelar uso" ao sistema `Sis` Fim se E5.3 Repete a partir de 1 Fim evento E5.

Exemplo: Caso de uso Efetuar Login

Pós condições	
Garantia mínima	
Requisitos	
Regras de negócio	
Casos de uso correlatos	

Exemplo: Caso de uso Efetuar Login

Pós condições	O sistema **Sis** recebeu o conjunto de direitos de uso correspondentes ao par <usuário, senha> fornecido
Garantia mínima	
Requisitos	
Regras de negócio	
Casos de uso correlatos	

Exemplo: Caso de uso Efetuar Login

Pós condições	O sistema **Sis** recebeu o conjunto de direitos de uso correspondentes ao par <usuário, senha> fornecido
Garantia mínima	O componente sempre retorna um par <condição de retorno, direitos de uso>, em que o conjunto de direitos de uso estará vazio.
Requisitos	
Regras de negócio	
Casos de uso correlatos	

Exemplo: Caso de uso Efetuar Login

Pós condições	O sistema **Sis** recebeu o conjunto de direitos de uso correspondentes ao par <usuário, senha> fornecido
Garantia mínima	O componente sempre retorna um par <condição de retorno, direitos de uso>, em que o conjunto de direitos de uso estará vazio.
Requisitos	• Segurança: o cadastro de usuários deve estar criptografado • Segurança: todos os espaços que contenham dados do cadastro e/ou do usuário devem ter sido obliterados antes de retornar • Segurança: a senha nunca deve ser exibida • Segurança: a cada ativação da entrada de dados, novos caracteres de controle devem ser fornecidos • Segurança: a cada ativação os campos são fornecidos em branco • IHC: usuários devem poder preencher os campos em qualquer ordem
Regras de negócio	
Casos de uso correlatos	

Exemplo: Caso de uso Efetuar Login

Pós condições	O sistema **Sis** recebeu o conjunto de direitos de uso correspondentes ao par <usuário, senha> fornecido
Garantia mínima	O componente sempre retorna um par <condição de retorno, direitos de uso>, em que o conjunto de direitos de uso estará vazio.
Requisitos	• Segurança: o cadastro de usuários deve estar criptografado • Segurança: todos os espaços que contenham dados do cadastro e/ou do usuário devem ter sido obliterados antes de retornar • Segurança: a senha nunca deve ser exibida • Segurança: a cada ativação da entrada de dados, novos caracteres de controle devem ser fornecidos • Segurança: a cada ativação os campos são fornecidos em branco
Regras de negócio	• A identificação de usuário e a senha não devem conter caracteres diacríticos, espaços em branco, tabulações, nem os caracteres '%', '\', '/', '?', '*', '@' • Os campos são sensíveis às caixa alta e baixa
Casos de uso correlatos	

Exemplo: Caso de uso Efetuar Login

Pós condições	O sistema **Sis** recebeu o conjunto de direitos de uso correspondentes ao par <usuário, senha> fornecido
Garantia mínima	O componente sempre retorna um par <condição de retorno, direitos de uso>, em que o conjunto de direitos de uso estará vazio.
Requisitos	• Segurança: o cadastro de usuários deve estar criptografado • Segurança: todos os espaços que contenham dados do cadastro e/ou do usuário devem ter sido obliterados antes de retornar • Segurança: a senha nunca deve ser exibida • Segurança: a cada ativação da entrada de dados, novos caracteres de controle devem ser fornecidos • Segurança: a cada ativação os campos são fornecidos em branco • IHC: usuários devem poder preencher os campos em qualquer ordem
Regras de negócio	• A identificação de usuário e a senha não devem conter caracteres diacríticos, espaços em branco, tabulações, nem os caracteres '%', '\', '/', '?', '*', '@' • Os campos são sensíveis às caixa alta e baixa
Casos de uso correlatos	Reconfirmar usuário corrente durante o uso de **Sis** Solicitar nova senha Mudar a senha Registrar dados de usuário Alterar dados de usuário Excluir dados de usuário

Formulário de especificação de casos de uso 1/3

Caso de uso	nome (identificação) do caso de uso	
Resumo	descrição resumida do objetivo principal	
Escopo	o que é abrangido pelo caso de uso	
Ator principal	nome do ator que ativa (aciona) o caso de uso	
Interessados	nome (identificação) do interessado	descrição do interesse / objetivo
Invariante	condições que deverão estar satisfeitas antes e após efetuar o caso de uso • devem envolver somente dados, arquivos, recursos e estados manipulados pelo caso de uso	
Pré condições	condições que precisam estar satisfeitas antes de iniciar o caso de uso • Invariante + Pré condições devem envolver todos os dados, arquivos, recursos e estados necessários para poder realizar todo o caso de uso	
Acionamento	condição ou evento que inicia o caso de uso, disparado pelo ator principal	

Formulário de especificação de casos de uso 2/3

Fluxo principal	sequência de ações a serem executadas no caso normal
Fluxos alternativos	sequências de ações disparados por eventos previstos durante a execução de ações do fluxo principal, ou de fluxos alternativos já descrito • fluxos alternativos devem indicar onde iniciam e onde terminam
Pós condições	condições que devem estar satisfeitas ao terminar a execução normal do caso de uso • invariante + pós-condições deve envolver tudo o que resulta ao final do fluxo normal • pós-condições devem poder ser utilizadas como o oráculo do teste
Garantia mínima	condições que devem ser satisfeitas sempre que o caso de uso não termine de forma normal. Possivelmente a invariante não estará totalmente assegurada ao terminar de forma não normal, portanto a garantia mínima deve relatar o todo o conjunto de condições asseguradas.

Formulário de especificação de casos de uso 2/3

Requisitos	requisitos adicionais, tais como requisitos não funcionais e características desejáveis da interface humano computador (IHC)
Regras de negócio	especificação das regras (assertivas) a serem satisfeitas nos diferentes passos
Casos de uso correlatos	relação dos casos de uso correlacionados com o presente caso de uso • ex. os demais casos de uso que complementam o conjunto de características que implementam um determinado propósito
Artefatos correlatos	relação de documentos, artigos, mensagens, fontes de informação, etc. que contêm alguma informação relevante para o caso de uso

Como redigir as ações

- As ações devem ser legíveis e compreensíveis por todos os interessados
  - muitos deles não têm, nem precisam ter, treinamento em Informática
    - a recomendação para a estrutura gramatical descritos a seguir foram inspirados em [Díaz et al, 2004] e [Heumann, 2001]

Como redigir as ações

- Solução: redigir em português restrito

  {<pré-condição> ":"} (<ator>|<entidade>) <atividade> {<restrições>}

  pré-condição – condição que deve estar satisfeita para que a atividade possa ser realizada
    - pode ser entendido como uma forma de guarda [Dijkstra, 1975]

  ator – papel (não a pessoa) que realiza ou dispara a atividade

  entidade – equipamento, sistema ou componente que realiza ou dispara a atividade

  atividade – ação a realizar, no formato:
    - <verbo> <objetos>
    - verbo na terceira pessoa do presente do indicativo

  restrições – condições a serem respeitadas ao realizar a atividade

Como redigir as ações

- Exemplos
  - O usuário digita a identificação, a senha, os caracteres de controle
  - O usuário seleciona a ação
  - Errado: o aluno foi para casa
  - Certo: o aluno vai para casa

- As atividades não devem ser compostas
  - Errado: o aluno vai para casa e estuda teste de software
  - Certo:
    - o aluno vai para casa
    - o aluno estuda teste de software

conector de expressão lógica

Como redigir as ações

Exemplo: sistema de controle de um elevador

- O <módulo controlador> <informa> ao <módulo motor de determinado poço> o <andar para o qual o elevador deve se deslocar>

ação sujeito verbo objeto indireto objeto direto

- O <módulo motor> calcula o <plano de movimentação>
  - O <plano de movimentação> é formado por: <direção> e <sequência de velocidades> que levam da <posição atual> ao <andar destino>
  - <posição atual> é o lugar em que o elevador se encontra no momento da consulta, pode ser entre andares
- O <módulo motor> executa o <plano de movimentação>

regra de negócio

informar – verbo bitransitivo, i.e. requer objeto direto **e** indireto

Como redigir os fluxos

- Redação do fluxo (*path*) principal
  1. {pré-condição :} (ator | entidade) atividade {restrições}
  2. {pré-condição :} (ator | entidade) atividade {restrições}

  ...

  n. {pré-condição :} (ator | entidade) atividade {restrições}
- Redação do fluxo alternativo

  Evento *i/j*: <condição identificadora do evento>

  E*i*.1. {pré-condição :} (ator | entidade) atividade {restrições}

  E*i*.2. {pré-condição :} (ator | entidade) atividade {restrições}

  ...

  E*i*.k. {pré-condição :} (ator | entidade) atividade {restrições}

  \/j é a linha em que ocorre o evento. Pode ser uma lista de linhas.
  - *j* será um número simples se o evento ocorrer no fluxo principal,
  - e será *Ei.j* se ocorrer em fluxo alternativo
  - eventos externos (ex. *time out*) não têm a parte */j*

Laboratório de Engenharia de Software

Como redigir controles

(Se | Quando | Dado) <condição> então

x.1. {pré-condição :} (ator | entidade) atividade {restrições}

...

Fim (se | quando | dado)

Para cada <elemento> (contido em | de) <conjunto> efetue

x.1. {pré-condição :} (ator | entidade) atividade {restrições}

...

Fim para cada

Laboratório de Engenharia de Software

Como redigir controles

Enquanto <condição> efetue

x.1. {pré-condição :} (ator | entidade) atividade {restrições}

...

Fim enquanto

Até <condição 1> efetue

x.1. {pré-condição :} (ator | entidade) atividade {restrições}

...

Após <condição 2> efetue

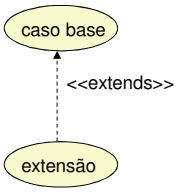
x.n. {pré-condição :} (ator | entidade) atividade {restrições}

...

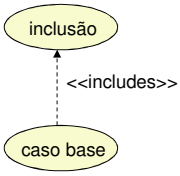
Fim até

opcional. Se existir deve ser a condição complemento da <condição 1>

Como redigir extensões e inclusões

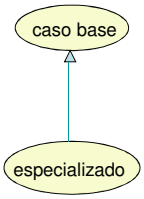


- estender <caso de extensão> para <caso base>
  quando <condição>
  {(antes | após) <ponto extensão>}



- incluir <caso de inclusão> em <caso base>
  {(antes | após) <ponto inclusão>}*

Como redigir especialização



- especializar <caso base> para <caso especializado>

Exemplo: Caso de uso Efetuar Login

fluxo principal	1. O componente limpa os campos e gera os caracteres de controle 2. O usuário digita sua identificação, senha e caracteres de controle 3. Quando o usuário selecionar a ação "OK" então 3.1 O controle de acesso verifica os dados fornecidos 3.2 O controle de acesso retorna ao sistema **Sis**, fornecendo a condição "autorizar uso" e os direitos de uso correspondentes a <usuário, senha> Fim quando 4. Quando o usuário selecionar a ação "Mudar senha" então 4.1 O controle de acesso verifica os dados fornecidos 4.2 O controle de acesso ativa o caso de uso "Trocar a senha" 4.3 Repete a partir de 1 Fim quando 5. Quando o usuário selecionar a ação "Esqueci a senha" então 5.1 O controle de acesso verifica os idUsuário e caracteres de controle 5.2 O controle de acesso ativa o caso de uso "Fornecer identificação alternativa" 5.3 Se retornar "cancela", repete a partir de 1 5.4 Senão retorna a **Sis** com <"Esqueceu" , vazio> Fim quando 6. Quando o usuário selecionar a ação "Cancelar" então 6.1 Se confirma retorna ao sistema **Sis**, fornecendo <"cancelar uso" , vazio > 6.2 Senão repete a partir de 1. Fim quando

Fluxo alternativo seria melhor?

A forma "... ativa ..." implica o uso de máquina de estados

Componente Login: especificação

fluxos alternativos	Evento 1/3.1,4.1: O usuário digitou identificação, senha ou caracteres de controle incorretos, ou o par <idUsuario, senha> não existe E1.1 Se for a quarta ou mais vez que ocorreu um evento de erro então E1.1.1. O controle de acesso emite a mensagem "Acesso não autorizado" E1.1.2. O controle de acesso retorna ao sistema **Sis**, fornecendo a condição "não autorizar uso" e direitos de uso nulo Fim se E1.2. Se caracteres de controle incorretos: o controle de acesso emite a mensagem "Caracteres de controle incorretos" E1.3. Se identificação do usuário ou senha incorretos: o controle de acesso emite a mensagem "Dados redigidos incorretamente" E1.4. Se o par <idUsuario, senha> não existe: o controle de acesso emite a mensagem "Usuário desconhecido" E1.5. O controle de acesso retorna ao passo 1 Fim evento

Incorreção: "O usuário digitou ... incorretos, ou o par <idUsuario, senha> não existe" deveria estar separado nas ações "O usuário digitou identificação, senha ou caracteres de controle incorretos", e outra ação "o par <idUsuario, senha> não existe"

Referências bibliográficas

Cockburn, A.; *Writing Effective Use Cases*; Reading, Massachusetts: Addison-Wesley; 2001

Cockburn, A.; *Escrevendo Casos de Uso Eficazes - Um Guia para Desenvolvedores de Software*; São Paulo, SP: Bookman; 2005

Díaz, I.; Losavio, F.; Matteo, A.; Pastor, O.; "A Specification Pattern for Use Cases"; *Information & Management* 41; New York, NY: Elsevier; 2004; pags 961-975

Dijkstra, E.W.; "Guarded commands, non-determinacy and formal derivation of programs"; *Communications of the ACM* 18(8); New York, NY: ACM Association for Computing Machinery; 1975; pags pp 453-457

Hall, A.; Chapman, R.; "Correctness by Construction: Developing a Commercial Secure System"; *IEEE Software* 19(1); Los Alamitos, CA: IEEE Computer Society; 2002; pags 18-25

Heumann, J.; "Generating Test Cases from Use Cases"; *The Rational Edge e-zine*; New York, NY: International Business Machines; 2001; Buscado em: 22/jan/2009; URL: www.ibm.com/developerworks/rational/library/content/RationalEdge/jun01/GeneratingTestCasesFromUseCasesJune01.pdf

Referências bibliográficas

Nebut, C.; Fleurey, F.; Jézéquel, J-M.; Traon, Y.L.; "Automatic Test Generation: A Use Case Driven Approach"; *IEEE Transactions on Software Engineering* 32(3); Los Alamitos, CA: IEEE Computer Society; 2006; pags 140-155

Pinto, T.D.; *Uma Ferramenta para Geração e Execução Automática de Testes Funcionais Baseados na Descrição Textual de Casos de Uso*; Dissertação de Mestrado, DI/PUC-Rio; 2013

o site da ferramenta é funtester.org

Stellman, A.; Greene, J.; 2014. *Learning Agile: Understanding Scrum, XP, Lean, and Kanban*. O'Reilly Media. Kindle Edition.

FIM