

# END-USER DEVELOPMENT (EUD) - DESENVOLVIMENTO PELO USUÁRIO FINAL: CONCEITOS, ESTRATÉGIAS E CASOS DE ADOÇÃO

## RELATÓRIO DE PESQUISA

Brasília, Agosto de 2017



UnB

MINISTÉRIO DA  
CIÊNCIA, TECNOLOGIA,  
INOVAÇÕES E COMUNICAÇÕES



E56 End-User Development (EUD): desenvolvimento pelo usuário final: conceitos, estratégias e casos de adoção : relatório de pesquisa / Rejane Figueiredo ... [et al.]. – Brasília : Editora Universidade de Brasília, Faculdade Gama, 2017. 71 p.

Modo de acesso: <http://repositorio.unb.br/handle/10482/24372>.

1. Informação tecnológica. 2. Sistemas de informação. 3. Desenvolvimento pelo usuário final. I. Figueiredo, Rejane.

CDU 004.414.2

*Preparado por*  
*ITRAC (Information Technology Research and Application Center) – unidade de pesquisa da*  
*Faculdade GAMA – FGA, Universidade de Brasília - UnB*  
*para o*  
*Ministério da Ciência, Tecnologia, Inovações e Comunicações - MCTIC*

**Autores:**

*Rejane Figueiredo (FGA/UnB, ITRAC)*

*Elaine Venson (FGA/UnB, ITRAC)*

*Claudia Melo (FT/UnB, ITRAC)*

*Carolina Ramalho (ITRAC)*

*Augusto Samuel Modesto (FGA/UnB, ITRAC)*

*Leonardo Sagmeister de Melo (FGA/UnB, ITRAC)*

*Allan Jeffrey Nobre (FGA/UnB, ITRAC)*

*Jonathan M. Moraes (FGA/UnB, ITRAC)*

*Marcelo Herton Pereira Ferreira (FGA/UnB, ITRAC)*

*Parley Martins (FGA/UnB, ITRAC)*

End-User Development (EUD) - desenvolvimento pelo usuário final: conceitos, estratégias e casos de adoção. Relatório técnico. FGA, UnB. Agosto, 2017.

*Pesquisa realizada com financiamento do Ministério da Ciência, Tecnologia, Inovações e Comunicações, Projeto de Cooperação “Aprimoramento do Framework de Soluções de Tecnologia da Informação”.*



## ÍNDICE

<b>1</b>	<b>INTRODUÇÃO</b>	<b>10</b>
<b>2</b>	<b>CONCEITOS</b>	<b>11</b>
2.1	DESENHANDO ORGANIZAÇÕES DE ALTA PERFORMANCE	11
2.2	TI CENTRALIZADA E TI DESCENTRALIZADA	12
2.3	DESENVOLVIMENTO DE SOFTWARE DESCENTRALIZADO ATRAVÉS DE END-USER DEVELOPMENT (EUD)	14
2.3.1	END-USER DEVELOPMENT (DESENVOLVIMENTO POR USUÁRIO-FINAL)	14
2.3.2	END-USER COMPUTING (COMPUTAÇÃO POR USUÁRIO FINAL)	15
2.3.3	RELAÇÃO ENTRE END-USER DEVELOPMENT E END-USER COMPUTING	15
2.4	ABORDAGENS DE DESENVOLVIMENTO POR USUÁRIOS FINAIS	16
2.5	PERFIL DO DESENVOLVEDOR USUÁRIO FINAL	18
2.6	TIPOS DE PROJETOS REALIZADOS POR USUÁRIOS FINAIS	20
2.7	CARACTERÍSTICAS, MOTIVAÇÕES, BENEFÍCIOS E RISCOS DE EUD	23
2.8	TENDÊNCIAS EM EUD E EUC	29
<b>3</b>	<b>ESTRATÉGIA DE ADOÇÃO DE DESENVOLVIMENTO</b>	<b>31</b>
3.1	TECNOLOGIAS E FERRAMENTAS EUD	34
3.2	MOTIVAÇÃO DOS USUÁRIOS	35
3.3	ANÁLISE DE ADOÇÃO DO EUD E AVALIAÇÕES DE APLICAÇÕES ADEQUADAS	36
3.4	IMPLANTAÇÃO E GESTÃO DO EUD	37
3.5	ANÁLISE DE IMPACTO	37
3.6	AVALIAÇÃO E MITIGAÇÃO DE RISCOS	38
<b>4</b>	<b>ANÁLISE DE ESTRATÉGIAS DE DESENVOLVIMENTO DESCENTRALIZADO EM INSTITUIÇÕES PÚBLICAS</b>	<b>40</b>
4.1	ESTUDO DE CASO A	40
4.1.1	HISTÓRICO DO EUD	40
4.1.2	ESTRATÉGIA DE ADOÇÃO DO EUD: ÁREAS DO NEGÓCIO, CRITÉRIOS PARA SELEÇÃO DAS SOLUÇÕES, SUPORTE DADO PELA TI	41
4.1.3	TECNOLOGIAS ADOTADAS	42
4.1.4	QUALIDADE E SEGURANÇA	42
4.1.5	IMPACTOS E BENEFÍCIOS	43
4.1.6	LIÇÕES APRENDIDAS	43
4.1.7	PERSPECTIVAS FUTURAS	44
4.1.8	APLICAÇÕES DESENVOLVIDAS NO MODELO DESCENTRALIZADO/EUD	44
4.2	ESTUDO DE CASO B	45
4.2.1	ESTRATÉGIA DE ADOÇÃO DO EUD: ÁREAS DO NEGÓCIO, CRITÉRIOS PARA SELEÇÃO DE SOLUÇÕES, SUPORTE DADO PELA TI	45
4.2.2	TECNOLOGIAS ADOTADAS	46

4.2.3	QUALIDADE E SEGURANÇA	47
4.2.4	IMPACTOS E BENEFÍCIOS	47
4.2.5	LIÇÕES APRENDIDAS	48
4.2.6	PERSPECTIVAS FUTURAS	48
<b>4.3</b>	<b>ESTUDO DE CASO C</b>	<b>48</b>
4.3.1	ESTRATÉGIA DE ADOÇÃO DO EUD: ÁREAS DO NEGÓCIO, CRITÉRIOS PARA SELEÇÃO DAS SOLUÇÕES, SUPORTE DADO PELA TI	49
4.3.2	TECNOLOGIAS ADOTADAS	50
4.3.3	QUALIDADE E SEGURANÇA	50
4.3.4	IMPACTOS E BENEFÍCIOS	50
4.3.5	LIÇÕES APRENDIDAS	51
4.3.6	PERSPECTIVAS FUTURAS	51
<b>4.4</b>	<b>ESTUDO DE CASO D</b>	<b>51</b>
4.4.1	HISTÓRICO DO EUD	51
4.4.2	ESTRATÉGIA DE ADOÇÃO DO EUD: ÁREAS DO NEGÓCIO, CRITÉRIOS PARA SELEÇÃO DAS SOLUÇÕES, SUPORTE DADO PELA TI	52
4.4.3	TECNOLOGIAS ADOTADAS	52
4.4.4	QUALIDADE E SEGURANÇA	53
4.4.5	IMPACTOS E BENEFÍCIOS	53
4.4.6	LIÇÕES APRENDIDAS	53
4.4.7	PERSPECTIVAS FUTURAS	53

<b>5</b>	<b><u>ANÁLISE DE ESTRATÉGIAS DE DESENVOLVIMENTO DESCENTRALIZADO EM EMPRESAS PRIVADAS</u></b>	<b>54</b>
<b>5.1</b>	<b>CASO UNION INVESTMENT E APEX</b>	<b>54</b>
5.1.1	HISTÓRICO	54
5.1.2	ESTRATÉGIAS	54
5.1.3	TECNOLOGIAS ADOTADAS	54
5.1.4	NECESSIDADES	55
5.1.5	SOLUÇÃO	55
<b>5.2</b>	<b>CASO LIBERTY SEGUROS E OUTSYSTEMS</b>	<b>56</b>
5.2.1	HISTÓRICO	56
5.2.2	ESTRATÉGIAS	56
5.2.3	TECNOLOGIAS ADOTADAS	57
5.2.4	NECESSIDADES	57
5.2.5	SOLUÇÃO	57
<b>6</b>	<b><u>REFERÊNCIAS</u></b>	<b>57</b>
<b>7</b>	<b><u>APÊNDICE A: PROCEDIMENTOS, PAPÉIS E RESPONSABILIDADES DO EUD NO ÓRGÃO PÚBLICO A</u></b>	<b>67</b>
<b>8</b>	<b><u>APÊNDICE B: CONCEITOS DE EUD E EUC</u></b>	<b>69</b>



## LISTA DE FIGURAS

<i>Figura 1 - O papel da TI dentro de uma corporação, adaptado (HODGKINSON, 1996).....</i>	<i>13</i>
<i>Figura 2 - Abordagem Sócio Técnica na Gestão do EUD. Fonte: Mehandjiev, Sutcliffe e Lee (2006).....</i>	<i>34</i>
<i>Figura 3 - Estratégias de desenvolvimento de soluções de TI do Órgão Público A .....</i>	<i>42</i>
<i>Figura 4 - Visão do consumo de dados dos sistemas centralizados e descentralizados do Órgão Público A .....</i>	<i>42</i>
<i>Figura 5 - Ciclo de vida do processo de desenvolvimento descentralizado do Órgão Público A.....</i>	<i>68</i>



## LISTA DE TABELAS

<i>Tabela 1 - Abordagens EUD, Spahn (2008).....</i>	<i>17</i>
<i>Tabela 2 - Características de Usuários-Finais (AMOROSO, 1992).....</i>	<i>19</i>
<i>Tabela 3- Matriz de Características de Usuário-Final - Amoroso (1992).....</i>	<i>20</i>
<i>Tabela 4 - Tipos de usuários finais, segundo Ko et. al (2011) .....</i>	<i>21</i>
<i>Tabela 5 - Principais características da Adoção do EUD .....</i>	<i>24</i>
<i>Tabela 6 - Principais motivações dos Usuário Finais .....</i>	<i>25</i>
<i>Tabela 7 - Principais benefícios na Adoção do EUD.....</i>	<i>27</i>
<i>Tabela 8 - Principais riscos para adoção do EUD .....</i>	<i>28</i>
<i>Tabela 9 – Exemplos de sistemas desenvolvidos no modelo descentralizado/EUD.....</i>	<i>44</i>
<i>Tabela 10 - Atividades e Papéis Responsáveis.....</i>	<i>67</i>
<i>Tabela 11 - Conceitos do EUD (end user development) no decorrer do tempo, baseado em Barker (2007) e Barker (2011) .....</i>	<i>69</i>
<i>Tabela 12 - Conceitos do EUC (end user computing), baseado em Barker (2007) e Barker (2011) .....</i>	<i>70</i>



# 1 INTRODUÇÃO

Uma das frentes de pesquisa e desenvolvimento do **Projeto P&D-MCTIC/UnB** (Projeto de Pesquisa e Desenvolvimento entre a Universidade de Brasília - UnB e o Ministério da Ciência, Tecnologia, Inovação e Comunicações - MCTIC), oriundo de termo de cooperação entre a UnB e o Ministério, teve como meta o estudo da adoção da Abordagem de Desenvolvimento pelo Usuário Final (EUD - *End-User Development*) como uma estratégia de desenvolvimento descentralizado de Tecnologia da Informação (TI) das organizações.

O EUD tem sido adotado como uma abordagem de descentralização da TI pelas organizações que possuem uma demanda alta por serviços de desenvolvimento de software/sistemas e que, na maioria das vezes, não conseguem corresponder às expectativas.

Normalmente, a área de TI dos órgãos públicos brasileiros possui uma capacidade limitada para atender toda a demanda de desenvolvimento de sistemas locais e departamentais. Algumas das causas envolvem:

- Critérios de priorização corporativo;
- Escassez de servidores públicos para gestão;
- Limitação orçamentária/financeira.

Muitas vezes, as áreas de negócio desenvolvem suas próprias soluções informacionais sem conhecimento da área de TI centralizada (CGTI - Coordenação Geral de TI) e sem processos ou metodologias que permitam a gestão e o apoio aos usuários desenvolvedores das áreas de negócio.

Paralelo a isso, nos últimos anos observa-se uma crescente disponibilização de ferramentas de software que empoderam os usuários, cada vez mais cientes e interessados no potencial que a tecnologia da informação pode trazer em seu trabalho.

Este trabalho é resultante de um estudo sobre as experiências de desenvolvimento descentralizado de sistemas, no qual as áreas negociais desenvolvem suas próprias soluções, com apoio, controle e gestão da área de tecnologia centralizada da organização.

Esta pesquisa teve como objetivo investigar cenários de uso e suas táticas de implementação, além de questões de tecnologia relacionadas à implementação e adoção efetivas de tecnologias EUD em contexto corporativo. O estudo engloba a análise de experiências do Desenvolvimento Descentralizado em órgãos públicos e empresas privadas abordando aspectos de i) Governança e Gestão; ii) Processos, e metodologias; iii) Pessoas; iv) Tecnologias e ferramentas adequadas ao Desenvolvimento Descentralizado.

Apresenta-se a pesquisa sobre Desenvolvimento pelo Usuário Final (*End-User Development*) como estratégia para a descentralização do desenvolvimento de sistemas em organizações. Este estudo servirá de subsídio para a tomada de decisão que vise à adoção de um modelo de desenvolvimento descentralizado para o MCTIC.

A pesquisa está organizada em 2 Relatórios. No primeiro relatório, parte 1- apresentam-se conceitos, estratégias de adoção e casos de adoção de órgãos públicos e de empresas privadas. No segundo relatório, parte 2 – apresenta-se uma análise de tecnologias / ferramentas disponíveis para emprego na abordagem EUD.

## 2 CONCEITOS

Nesta seção, apresentam-se os principais conceitos e elementos relacionados ao desenvolvimento descentralizado no século 21.

### 2.1 Desenhando organizações de alta performance

Estruturar uma organização não é uma tarefa simples, nem rápida. Organizações são vivas, mudam e evoluem continuamente. A estrutura organizacional deve propiciar a melhor organização possível para que uma organização entregue o valor que se propõe. Estruturas são feitas de diversos trade-off, forças contraditórias que precisam ser constantemente harmonizadas. Exemplo disso são as forças da inovação e da eficiência, elementos que todas as organizações precisam, mas cuja natureza de atividades envolvidas e prioridades é bastante diferente.

Mintzberg (1980) classificou diversos tipos de configuração de uma estrutura organizacional, desde as estruturas simples até as adhocracias. Cada configuração é mais adequada para certo tipo de trabalho, do valor a ser entregue.

Por exemplo, se uma organização quer ser altamente inovadora, precisa ter uma configuração com maior descentralização de decisões e alta fluidez que permita as pessoas se organizarem de acordo com iniciativas e ideias. Essa estrutura aproxima-se bastante de uma **adhocracia**.

Por outro lado, se uma organização existe em um ambiente estável, com poucas mudanças e, portanto, poucas diferenças de tomada de decisão, pode estruturar-se com maior centralização das decisões e baixa flexibilidade de reorganização, visto que ela não será necessária. Muitas vezes, essas organizações também são influenciadas por controles externos, figuras como um CEO ou outra empresa/órgão controlador, detentor do conhecimento e do poder para definir o melhor para toda a organização. Essa organização é uma **máquina burocrática**.

As máquinas burocráticas são uma ótima opção para a produção em massa, onde produtos, processos e sistemas de distribuição são geralmente racionalizados e, portanto, fáceis de compreender. Essa configuração é muito comum entre empresas de produção em massa grandes e maduras, como os fabricantes de automóveis, companhias de seguros e ferrovias. O McDonald's é um exemplo clássico dessa configuração - alcançando enorme sucesso em sua indústria simples através de meticulosa padronização.

Como os controles externos estimulam a burocratização e a centralização, essa configuração é muitas vezes assumida por organizações que são fortemente controladas do exterior. É por isso que as agências governamentais, que estão sujeitas a muitos desses controles, tendem a ser levadas a adotarem uma configuração de máquina burocrática, independente de suas outras necessidades e objetivos (MINTZBERG, 1981).

Mintzberg (1981) também descreve os lendários problemas da máquina burocrática: o trabalho maçante e repetitivo, empregados alienados, obsessão pelo controle (tanto dos mercados quanto dos trabalhadores), tamanho maciço e inadaptação. Tratam-se de máquinas adaptadas a finalidades específicas, não à adaptação a novas finalidades.

Assim, é importante observar que a **configuração da organização é fundamental para o cumprimento de sua missão**. Uma configuração inapropriada jamais levará a organização para o objetivo traçado. Além disso, o mecanismo de tomada de decisão é parte importante da estrutura organizacional.

## 2.2 TI centralizada e TI descentralizada

Dentro do contexto de configuração da estrutura organização, há uma discussão sobre centralização e descentralização de funções. No entanto, muitos líderes ainda não perceberam que esses são dois extremos que raramente atendem todas as nuances e necessidades específicas de certa organização (CRAMM, 2008). Uma importante pergunta é "Como descentralizar", bem mais do que "Centralizar ou Descentralizar".

No contexto da Tecnologia da Informação (TI), a Figura 1 mostra os extremos do papel da TI em uma organização, desde a radicalmente centralizada até a radicalmente descentralizada.

Em um extremo, a função de TI totalmente centralizada leva à total ausência de ownership dos sistemas pelas unidades de negócio demandantes. Elas também não têm controle dos custos envolvidos. A chance de se atingir os objetivos das unidades de negócios também é menor, tanto pela distância do domínio quanto pela dificuldade em atender as unidades de negócio em um tempo razoável.

Quando a função de TI é descentralizada, os custos gerais para cada grupo aumentam excessivamente. Há mais reinvenção da roda e menor sinergia no que tange ao desenvolvimento de soluções tecnológicas. Há também grande variação das competências dos sistemas de informação entre cada unidade.

Nesse *continuum*, há diferentes possibilidades de centralização/descentralização do poder de decisão. No centro da Figura 1, apresentam-se as possibilidades de meio termo, uma estrutura conhecida como Federada. Nessa estrutura busca-se capturar todos os benefícios da centralização a descentralização da função de TI.

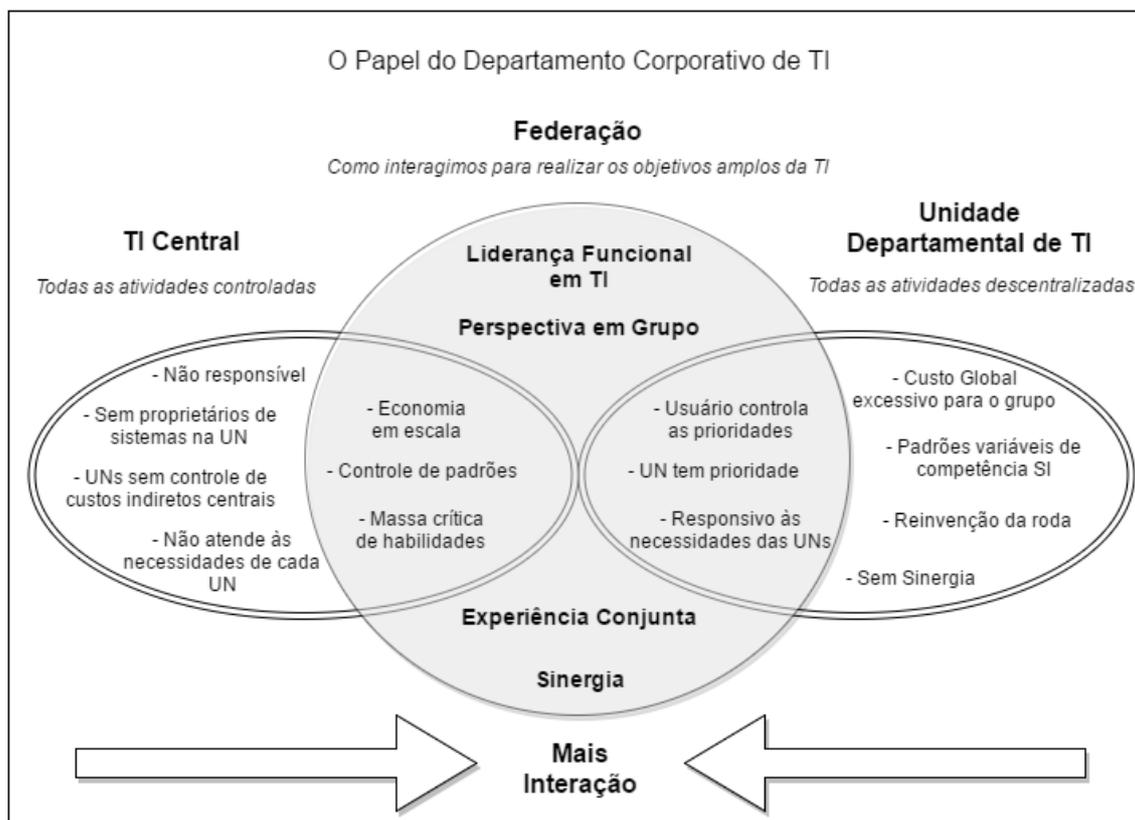


Figura 1- O papel da TI dentro de uma corporação, adaptado (HODGKINSON, 1996)

Nesse contexto, há ainda de se considerar que os próprios usuários possam ser os desenvolvedores de algumas das soluções tecnológicas. Em uma síntese do estado da arte em EUD e Engenharia de Software do Usuário-Final (*End-User Software Engineering - EUSE*), Ko et al. (2011) concluem que **a maior parte dos programas hoje não são escritos por profissionais de desenvolvimento de software, mas pelas pessoas com conhecimento de negócio** que precisam cumprir objetivos através da computação. Os conceitos de EUD e EUSE são apresentados nas próximas seções.

A atividade de desenvolver soluções tecnológicas de maneira autônoma, mesmo sem formação aprofundada, é uma realidade, visto a proliferação dos movimentos "faça você mesmo". Inclusive, a inovação, chave da Economia Digital, depende dessa rápida apropriação tecnológica por parte de usuários comuns, dado que a inovação pode surgir de qualquer um.

Devido à facilidade de uso e comoditização da tecnologia, os usuários finais tendem a ficar cada vez mais competentes em tecnologia. Eles querem tecnologias que os ajudarão a ter acesso a informações disponibilizadas em tempo real na Internet e Intranet de suas organizações, e processá-las sem ter que esperar que desenvolvedores criem programas e/ou relatórios adequados (LEWIS, 2010). Esse **fenômeno de empoderamento dos usuários** é

irreversível e tendem a aumentar exponencialmente, mesma curva de desenvolvimento tecnológico que se vivencia desde a constatação da Lei de Moore.

Portanto, além de definir o grau de descentralização para que a organização possa atingir seus objetivos e o locus da tomada de decisão, é preciso compreender as competências em tecnologia existentes na organização, ou mesmo o desejo de se desenvolver tais competências. Os avanços tecnológicos são disponibilizados diariamente, fenômeno que move a atual era da Economia Digital.

Independente da solução final, o objetivo mantém-se o mesmo: aumentar a efetividade e a eficiência da organização na realização de sua missão.

## 2.3 Desenvolvimento de software descentralizado através de End-User Development (EUD)

Há muitos sinônimos para o desenvolvimento de software descentralizado, como **desenvolvimento departamental, local ou setorial**.

Nesta pesquisa, observou-se também o uso frequente do termo *Shadow IT* para representar todas as soluções tecnológicas desenvolvidas de maneira autônoma, fora da gestão do departamento de TI. A diferença fundamental entre Shadow IT e EUD é que na última, as iniciativas são conhecidas pela organização e recebem suporte para ocorrerem (ZIMMERMANN et al., 2014). Independente no nível de suporte, ambas se baseiam na mesma premissa: o reconhecimento do valor inovativo e econômico que o desenvolvimento dirigido pelo usuário pode trazer.

Outras abordagens similares a EUD são **Rapid Application Development (RAD)** e **Low Code**. Esse último termo sendo cunhado recentemente pela indústria para caracterizar plataformas que permitem a entrega rápida de aplicativos de negócios com um mínimo de codificação manual e Investimento inicial de configuração, treinamento e implantação (RICHARDSON et al., 2016).

Nas próximas seções, definimos os conceitos centrais de *End-User Development* (Desenvolvimento por Usuário-Final) e *End-User Computing* (Computação por Usuário-Final), além do perfil do usuário final e tipos de sistemas desenvolvidos.

### 2.3.1 End-User Development (Desenvolvimento por Usuário-Final)

O End-User Development (EUD), ou Desenvolvimento por Usuário Final, visa capacitar os usuários finais e não-especialistas em programação de aplicações a desenvolver e adaptar sistemas de acordo com suas necessidades, sejam profissionais, educacionais ou para lazer (LIEBERMAN et al., 2006). Do ponto de vista da Engenharia de Software, EUD significa, em geral, a “participação ativa dos usuários finais no processo de desenvolvimento de software” (COSTABILE et al., 2005).

O Apêndice D detalha as diversas definições de EUD encontradas na literatura, uma área de conhecimento abrangente e não completamente consolidada.

### **2.3.2 End-User Computing (Computação por Usuário Final)**

O End-User Computing (EUC), ou Computação por usuário Final, envolve a interação de gestores, profissionais e usuários de nível operacional com aplicações software dentro de seus próprios departamentos de trabalho (BARKER, 2007). O Apêndice D detalha as diversas definições de EUC encontradas na literatura.

### **2.3.3 Relação entre End-User Development e End-User Computing**

Apesar dos conceitos do end-user computing e do end-user development, serem bastante similares, Barker (2007) ressalta:

*EUC pode agora ser definido como a utilização de tecnologias computacionais ou aplicações de software, juntamente com a otimização ou desenvolvimento de sistemas da informação pelos usuários finais. Por outro lado, o EUD é, mais especificamente, o desenvolvimento, modificação ou aperfeiçoamento de aplicações de sistemas da informação por end-users para uso individual, departamental ou organizacional.*

Em outras palavras, o EUC engloba o EUD no sentido de framework e ambiente computacional, que, realmente só surgiu em meados de 1980, devido ao surgimento dos computadores pessoais. Contudo o primeiro, sem suporte e sem notoriedade empresarial, apesar de poder desenvolver de diversas formas e em vários níveis, trabalha somente para tarefas individuais.

Ao passo que, com o crescimento do EUC, as empresas desenvolvem estratégias para aumentar os rendimentos internos individuais, que fez com que o EUD emergisse.

Assim, o EUD foi capaz de realizar todas as tarefas do EUC, contudo, com capacidade e preparação - fornecidas pela administração da organização - para dar suporte não somente ao desenvolvimento, modificação ou aperfeiçoamento de aplicações individuais, mas também departamentais e organizacionais, aumentando a eficácia e a eficiência da resolução de tarefas empresariais.

Segundo Barker (2007), o EUD é uma parte importante do EUC e além de poder apoiar o usuário-final desenvolvedor (*end-user developer*) em suas tarefas diárias, apoia as funções de outros usuários finais no departamento.

## 2.4 Abordagens de Desenvolvimento por usuários Finais

Existem diversas formas de se implementar o desenvolvimento por usuários-finais, cada uma com características únicas foram desenvolvidas, apropriadas para diferentes situações. Algumas exigem níveis de conhecimento técnico superiores a outras, dependendo de sua implementação. A partir disto, Lieberman et al. (2006) descrevem a importância de uma curva de aprendizado de EUD por meio da utilização de diversas abordagens com diferentes níveis de complexidade. A mesma posição é defendida por Spahn (2008).

Para auxiliar no processo de compreensão das abordagens, Spahn (2008) desenvolveu uma tabela de caracterização segundo o nível de conhecimento exigido e o nível de adaptabilidade oferecido. Ela é organizada segundo duas grandezas: Complexidade e Poder de Adaptabilidade.

**Complexidade** diz respeito ao nível de conhecimento técnico necessário para se utilizar a metodologia, e está dividida em Programadores (para o mais alto nível), Desenvolvedores Locais (para aqueles que possuem certo conhecimento, mas não desenvolvem aplicações por si só) e Não-Programadores (para aqueles que não possuem nenhum conhecimento técnico em programação). Estes níveis surgiram a partir da classificação de usuários-finais de Nardi e Miller (1990).

**Poder de adaptação** diz respeito ao nível de adaptação que a abordagem oferece à aplicação, e está organizada em Personalização (capacidade de alterar a aparência e comportamento da aplicação de forma pré-definidas e fechadas), Integração (capacidade de adaptar o desenho de aplicações a partir de modelos) e Extensão (capacidade de modificar a aplicação por meio de código). Esses níveis surgiram a partir da classificação de Mørch (1997). Na Tabela 1 apresentam-se as diferentes abordagens de EUD.

Além dessas grandezas, na Tabela 1 apresenta-se uma seção de abordagens de suporte ao EUD, tais como o teste baseado em questões e o princípio, “o que se vê é o que se testa” (*What You See Is What You Test*). Essas abordagens servem para auxiliar as principais apresentadas na Tabela.

O nível mais baixo de complexidade é o apresentado para **Não-Programadores**. Os não-programadores podem adaptar aplicações de software a partir de:

- Personalização
  - o **Personalização de Interface:** forma simples de adaptar um sistema às necessidades de um usuário, oferecendo apenas a possibilidade de alterar a aparência da interface de aplicações. Sendo um nível tão básico, é uma abordagem amplamente utilizada atualmente.
  - o **Parametrização:** possibilidade de alterar certas configurações da aplicação a partir de um conjunto de opções pré-definido.
- Integração

- o **Programação por Demonstração:** envolve um sistema no qual o usuário utilize o computador de forma usual, e o mesmo seria capaz de notar padrões de uso e replicá-los futuramente de forma autônoma. É utilizado em ferramentas como o Microsoft Excel.
- o **Paradigma dos Contadores:** é o formato utilizado por planilhas, no qual os dados são apresentados em tabelas que podem ter seu comportamento definido por meio de equações, por exemplo.
- o **Interface de Ajuste Integrada:** pode normalmente ser encontrada em conjunto com a anterior, pois diz respeito à união entre as visões de adaptação e uso em uma visão única, assim como se observa em planilhas.

Abordagens EUD			
Poder de Adaptabilidade	Personalização	Integração	Extensão
Complexidade			
Programadores(as)			Programação
Desenvolvedores(as) Locais		Troca de componentes a tempo de execução; Interface de ajuste separada	Programação natural; Programação de script
Não-programadores(as)	Personalização de interface; Parametrização	Programação por demonstração; Paradigma dos contadores; Interface de ajuste integrada	

Tabela 1- Abordagens EUD, Spahn (2008)

Em um nível intermediário de complexidade, tem-se o Desenvolvedor Local. O desenvolvedor-local pode adaptar aplicações de software a partir de:

- Integração
  - o **Troca de Componentes em Tempo de Uso:** traz a ideia de o usuário de um sistema baseado em componentes trocar os mesmos durante o uso (i.e., sem ter de reiniciar a aplicação para realizar as alterações desejadas).
  - o **Interfaces de Tailoring Separadas:** semelhante à abordagem de interfaces de tailoring integradas, sua única diferença é a separação entre as interfaces de

uso e adaptação. Isto proporciona uma gama maior de possibilidades de edição, apesar de adicionar complexidade ao uso.

- Extensão
  - o **Programação Natural e Programação de Script:** não exigem um nível de conhecimento técnico tão alto, se tratando da utilização de linguagens naturais de fácil entendimento por não-especialistas e de linguagens de script, que são mais simples e menos poderosas do que linguagens de programação propriamente ditas.

Por fim, no mais alto nível de complexidade e adaptabilidade, está a utilização de programação formal para alterar aplicações e executar as adaptações desejadas. Nesse caso, apenas programadores têm o conhecimento necessário para realizar esse tipo de tarefa.

## 2.5 Perfil do Desenvolvedor Usuário Final

Uma das classificações de usuários-finais é a taxonomia proposta por Rockart e Flannery (1983). Nela definem-se seis perfis distintos, divididos pelos fatores i) nível de conhecimento de computação, ii) nível de assistência que requerem e recebem e iii) modo como utilizam processos computadorizados de manipulação de informação.

Segundo a taxonomia de Rockart e Flannery (1983), os end-users dividem-se em:

- **Usuário-Final não-programador (Non-programming end-user):** usuário mais comuns de sistemas de informação, utilizam sistemas criados por terceiros e não possuem participação alguma no processo de desenvolvimento;
- **Usuário-Final de nível de comando (Command level end-users):** usuário com o nível mais básico de envolvimento no processo de computação, chegam a aprender um mínimo necessário para manusear dados (e.g. criar queries em banco de dados);
- **Usuário-Final programador (End-user programmers):** usuários que possuem um alto nível de especialização em sua área assim com algum conhecimento de programação, de forma a ser capaz de desenvolver aplicações simples para seu próprio uso, ou de outros em seu departamento;
- **Pessoal de suporte funcional (Functional support personnel):** usuário que, diferentemente do Usuário-Final programador, possui baixo nível de especialização em um departamento, mas possui grande conhecimento acerca do software utilizado, e irá auxiliar o Usuário-Final programador;

- **Pessoal de suporte de EUC (End-user-computing support personnel):** usuário responsável por fornecer o suporte necessário. Possui conhecimento geral de diversas técnicas de programação;
- **Programadores processadores de dados (Data processing programmers):** usuário especialista que possui pouco conhecimento contextual e muito conhecimento da aplicação ou sistema utilizado. Responsável por desenvolver aplicações especializadas não triviais.

Amoroso (1992) observa dez características distintas para usuários-finais. Ele define escalas qualitativas para cada uma delas, como mostra a Tabela 2.

*Tabela 2 - Características de Usuários-Finais (AMOROSO, 1992)*

<b>Características do Perfil</b>	<b>Escalas possíveis</b>
<b>Habilidades de computação</b> ( <i>computing skills</i> )	Baixa vs. Alta
<b>Programação exigida</b> ( <i>programming required</i> )	Para si vs. Para outros
<b>Nível de suporte de EUC</b> ( <i>level of EUC support</i> )	Exigido por vs. Fornecido por
<b>Educação e treinamento</b> ( <i>training and education</i> )	Poucas ferramentas vs. Múltiplas ferramentas
<b>Conhecimento de ferramentas de EUC</b> ( <i>knowledge of EUC tools</i> )	Manipulação vs. Desenvolvimento
<b>Localização de usuários-finais</b> ( <i>location of end-users</i> )	Funcional vs. Grupo de IS (sistemas de informação)
<b>Natureza das aplicações</b> ( <i>nature of applications</i> )	Pequenas/Simples vs. Grandes/Complexas
<b>Compreensão técnica</b> ( <i>technical understanding</i> )	Principiante vs. Experiente

<b>Atitudes de usuário-final</b> (end-user attitudes)	Positivas vs. Negativas
<b>Experiência de usuário-final</b> (end-user experience)	Baixa vs. Alta

Amoroso ainda analisa as características para cada categoria definida por Rockart e Flannery (1983). Esse cruzamento é apresentado em uma matriz características versus categorias, como mostrado na Tabela 3.

## 2.6 Tipos de projetos realizados por usuários finais

Klann et al (2006) definem algumas áreas nas quais o EUD pode ser ou vem sendo aplicado nos últimos anos. O primeiro exemplo é o **uso doméstico**. Como tem sido observado, o contexto dos sistemas de software tem estado cada vez mais presente no dia a dia da população, por meio de diferentes formas. Os autores discutem sobre a tendência destes sistemas se tornarem cada vez mais flexíveis às necessidades específicas de cada usuário, podendo ser facilmente adaptadas e interconectadas umas às outras.

No contexto de aplicações visuais, há o exemplo do **design industrial** com sistemas CAD. Klann et al (2006) descrevem como designers se beneficiam do EUD criando ou modificando o próprio ambiente de construção de forma a tornar o mesmo ideal para suas necessidades. Além disso, esses designers se tornam especialistas em tais ambientes, uma vez que foram os responsáveis por sua criação.

Os autores trazem ainda um exemplo na área de **pesquisa científica**. Mais especificamente, mencionam a existência de diversas aplicações criadas pelo departamento de biologia do Instituto Pasteur em Paris, criadas para auxiliar pesquisadores a tratar e comparar diversos tipos de dados. Este tipo de aplicação pode ser uma vantagem em diversas áreas de investigação científica, como na física, química ou na medicina.

*Tabela 3- Matriz de Características de Usuário-Final - Amoroso (1992)*

Tipos de End-User/Características	End-User Não-Programador	End-User de Nível de Comando	End-User Programador	Pessoal de Suporte Funcional	Pessoal de Suporte de EUC	Programadores Processadores de Dados
Habilidades de Computação	Pouco	Baixo  Alto				
Programação Exigida	Nenhuma	Para si		Para outros		
Nível de Suporte de EUC	Exigido por			Fornecido por		
Educação e Treinamento	Poucas ferramentas  Muitas ferramentas					
Conhecimento de Ferramentas de EUC	Manipulação	Manipulação / Desenvolvimento		Desenvolvimento		
Localização de End-Users	Área funcional				Centro de Informação	Grupo de IS
Natureza das Aplicações	Pequenas/Simples  Grandes/Complexas					
Compreensão Técnica	Principiante  Experiente					
Atividades de End-User	"Deixe que outros façam"	"Eu mesmo faço"		"Deixe-me ajudar"		
Experiência de End-User	Pouca	Baixa  Alta				

Além destes exemplos, Rittenberg (1993) ainda cita brevemente os contextos de **aplicações financeiras, administrativas, de recursos humanos**, dentre outros. Ele enumera diversos exemplos de aplicações como geração de formulários, estimação e gerenciamento de aposentadoria, análise de inventário, análise e rastreamento de consumo de energia e sistemas de gerenciamento de projetos.

Um levantamento do estado da arte no emprego de técnicas da Engenharia de Software no EUD foi realizado por Ko et. al (2011). Eles reuniram um conjunto de atividades, ferramentas e linguagens que podem ser utilizadas para o desenvolvimento de soluções de software por diferentes tipos de usuários finais. Na Tabela 4 apresentam-se os tipos de usuários finais e as atividades, ferramentas e linguagens recorrentemente utilizadas por eles.

*Tabela 4 - Tipos de usuários finais, segundo Ko et. al (2011)*

Tipos de Usuário Final	Atividades, Ferramentas e Linguagens utilizadas pelo Usuário Final
<b>Administradores de Sistemas</b>	Escrever <i>scripts</i> para unir sistemas usando editores de texto e linguagens de <i>script</i>
<b>Designers de Interação</b>	Criar protótipo de interfaces de usuário com ferramentas

	como <i>Visual Basic</i> e <i>Flash</i>
<b>Artistas</b>	Criar arte interativa com linguagens como <i>Processing</i> ( <a href="http://processing.org">http://processing.org</a> )
<b>Professores</b>	Ensinar ciência e matemática com planilhas eletrônicas (NIESS et al. 2007)
<b>Contadores</b>	Tabular e resumir dados financeiros com planilhas
<b>Estatísticos</b>	Calcular e avaliar riscos usando ferramentas de simulação financeira como MATLAB
<b>Arquitetos</b>	Modelar e estruturas de projeto usando <i>FormZ</i> e outros modeladores 3D
<b>Crianças</b>	Criar animações e jogos com <i>Alice</i> (DANN et al. 2006) e <i>Scratch</i>
<b>Garotas do Ensino Médio (ou Estudantes)</b>	Usar <i>Alice</i> para contar histórias (KELLEHER e PAUSCH, 2006; KELLEHER e PAUSCH, 2007)
<b>Webmasters</b>	Gerenciar bancos de dados e sites usando o <i>Access</i> , <i>FrontPage</i> , <i>HTML</i> , <i>Javascript</i>
<b>Profissionais de Saúde</b>	Escrever especificações para gerar formulários de relatório médico
<b>Cientistas e Engenheiros</b>	Usar <i>MATLAB</i> e <i>Prograph</i> para realizar testes e simulações
<b>Usuários de E-mail</b>	Escrever regras de e-mail para gerenciar, classificar e filtrar e-mails
<b>Jogadores de Vídeo Game</b>	Automatizar "mods" para atiradores em primeira pessoa, jogos multiplayer online e <i>The Sims</i>
<b>Músicos</b>	Criar música digital com sintetizadores e idiomas de fluxo de dados musicais
<b>Usuários do VCR e TiVo (Televisores)</b>	Gravar programas de televisão com antecedência, especificando parâmetros e horários

<b>Donas de Casa</b>	Escrever programações de controle para sistemas de aquecimento e iluminação com <i>X10</i>
<b>Usuários do Apple OS X</b>	Automatizar o fluxo de trabalho usando o <i>AppleScript</i> e o <i>Automator</i>
<b>Usuários de Calculadora</b>	Processar e gerar gráficos com dados matemáticos a partir de linguagens de <i>script</i> de calculadora
<b>Gerentes</b>	Automatizar e gerar relatórios baseados na base de dados com o <i>Crystal Reports</i>

Nota-se que as atividades dos usuários finais são distintas de acordo com o tipo de usuário final, isto é, conforme o objetivo que o usuário final deseja alcançar, o conjunto de atividades e ferramentas utilizados para atingir este objetivo são diferentes. Por exemplo, os tipos de usuários Contadores e Usuários de Calculadora podem ser assumidos simultaneamente pela mesma pessoa.

No entanto, as atividades e ferramentas que serão utilizadas para o desenvolvimento das soluções de software são diferentes, ora tabular e resumir dados financeiros com planilhas ora processar e gerar gráficos com dados matemáticos a partir de linguagens de script de calculadora, conforme os objetivos dos tipos de usuários.

## 2.7 Características, Motivações, Benefícios e Riscos de EUD

O desenvolvimento de software pelo usuário final (EUD) é uma estratégia que pode diminuir a volumosa demanda da área de negócio para TI. Sobretudo, muitos benefícios são encontrados na literatura após a adoção da estratégia EUD como desenvolvimento descentralizado de software, por exemplo a satisfação do cliente em relação a TI. Mas, para que a TI e a área de negócio possam aproveitar dos benefícios da adoção do EUD, algumas características devem ser observadas e analisadas para que haja uma constante motivação dos usuários finais para desenvolver suas aplicações (conhecidas como UDA - *User Developed Applications*).

Foi realizado um levantamento das principais características, motivações, benefícios e riscos na literatura. As publicações foram selecionadas a partir de uma revisão bibliográfica utilizando o termo “*End User Development*” e algumas variações como “EUD”, “*End User Computing*”, “EUC” entre outros.

Um estudo realizado acerca do End-User-Development levantou 4 principais tópicos: características, motivações, benefícios e riscos. As características envolvem as

particularidades que um EUD pode possuir. Na Tabela 5 apresentam-se as principais características identificadas”.

*Tabela 5 - Principais características da Adoção do EUD*

<b>Autores</b>	<b>Características</b>	<b>Considerações</b>
<b>Panko &amp; Halverson (1996)</b>	Pouca ou nenhuma auditoria no uso e desenvolvimento de UDA (Aplicações Desenvolvidas pelo Usuário Final)	Organizações realizam poucas avaliações formais de adequação ao uso e do desenvolvimento
<b>McGill (2004); Lin &amp; Shao (2000); Doll and Torkzadeh (1988)</b>	Quanto maior a participação do usuário final no desenvolvimento maior os benefícios	Dentre os benefícios estão satisfação com o processo e com a UDA e maior conhecimento da aplicação
<b>Huitfeldt e Middleton (2001)</b>	Usuários Finais não sabem avaliar a qualidade de suas aplicações	Deve-se adotar estratégias como padrões, processos, políticas, dentre outros.
<b>Boehm et al (1978); Cavano &amp; McCall (1978); Amoroso e Cheney (1992)</b>	Usuários Finais têm dificuldades em reconhecer a qualidade de suas UDAs	Pela ausência de conhecimento técnico
<b>McGill (2004)</b>	Utilizar um processo bem definido proporciona bons ganhos	Utilizar um processo fornecerá preparação para uso futuro de uma aplicação, reduzir a dependência de treinamentos e documentação.
	Usuário Finais depende de documentação e	Quando o usuário participa das atividades de

treinamento caso não participe do desenvolvimento	desenvolvimento, ele desenvolve habilidades para o manuseio de sua UDA, quando o usuário não participa, ele terá dificuldades para aprender
Utilizar a própria UDA gera mais percepção dos benefícios do que utilizar de outros desenvolvedores	Utilizar UDA de outros usuários finais requer um esforço de aprendizagem do sistema, mas não gera a sensação de satisfação própria
Artefatos podem ser difíceis de auditar	A natureza do artefato pode impossibilitar a automatização da auditoria. Uma auditoria manual pode custar esforços da TI
Utilizar políticas de acesso a dados de acordo com o domínio é importante	Existem domínios com restrições de acesso a dados, neste caso, EUD não é recomendado

As motivações envolvem os ensejos responsáveis pela conversão de um usuário comum em um EUD. Na Tabela 6 apresenta-se motivações que levam os usuários a participarem do EUD, bem como algumas estratégias para motivá-los durante o desenvolvimento de aplicações.

*Tabela 6 - Principais motivações dos Usuário Finais*

<b>Autores</b>	<b>Motivações</b>	<b>Considerações</b>
<b>Barki &amp; Hartwick (1994); Doll &amp; Torkzadeh (1989)</b>	Maior sensação de envolvimento e participação	O usuário final acreditar que sua participação influencia no resultado

		final
<b>Barki &amp; Hartwick (1994); McGill (2004); Cheney, Mann e Amoroso (1986)</b>	Maior comprometimento para sucesso o UDA	O usuário se envolve com mais responsabilidade
<b>Cheney, Mann e Amoroso (1986)</b>	Maior compreensão do problema	O usuário sabe melhor do que ninguém o que ele precisa, mas tem dificuldade de expressar
<b>Mumford (1981); Nardi (1993); Mehandjiev et al (2006)</b>	Utilização de comunidades de apoio ao usuário-final	Usuários puderam se informar e tirar suas principais dúvidas, podendo interagir com uma comunidade de outros usuários finais
	Tornar os benefícios e histórias de sucesso visíveis aos usuários-finais	Permite que outros usuários finais se motivem
	Incentivar o envolvimento do usuário no desenvolvimento	Quanto mais motivado, mais chances de sucesso
<b>Mehandjiev et al (2006)</b>	Incentivar o compartilhamento de conhecimento e soluções	Permite que outros usuários finais se motivem
	Reduzir os custos de aprendizagem	Garantindo aos usuários finais tempo para aprender novas ferramentas e participar de sessões de treinamento

Os benefícios envolvem os ganhos pelos quais se avaliar que é melhor para a organização ter usuários-finais desenvolvedores em vez de ter apenas usuários comuns e uma

TI centralizada, trabalhando de forma separada. Ao todo foram elicitados dez benefícios, expostos na Tabela 7.

*Tabela 7 - Principais benefícios na Adoção do EUD*

Autores	Benefícios	Considerações
<b>Brancheau &amp; Brown (1993); Shayo, Guthrie, &amp; Igbaria (1999)</b>	Maior oportunidade de acesso à informação	Usuário finais podem ter acesso a novos ambientes, conhecimentos, práticas e acesso a dados
	Melhora a qualidade da informação	Maior entendimento da utilidade da informação
	Melhora a tomada de decisão	Quando o usuário participa do processo de desenvolvimento ele tem mais competência para julgar decisões
	Reduz o atraso de desenvolvimento de aplicações solicitadas a TI	Desenvolve software menos priorizados pela TI
<b>McGill (2004)</b>	Reduz a sobrecarga na TI	Permite que a TI se preocupe com software de grande porte (mais complexos)
	Melhora a relação do pessoal da TI e Usuários Finais	Melhora a comunicação e entendimento das demandas de um para o outro
<b>Lin &amp; Shao (2000)</b>	Aumenta o número de uso de aplicações	Quando o usuário desenvolve sua própria aplicação, ele valoriza mais o produto

Maior aceitação da UDA pelo usuário

<b>Lin &amp; Shao (2000); McGill (2004); Doll &amp; Torkzadeh (1989); McGill et al. (1998)</b>	Satisfação geral do usuário	Quanto a TI e quanto a UDA, processos, dentre outros
<b>McGill (2004); Amoroso (1988)</b>	Melhora o desempenho do usuário ao utilizar a UDA	Quando o usuário participou do desenvolvimento

Os riscos envolvem as intempéries que um EUD pode trazer para a organização, de forma indesejada, não monitorada ou não prevista. Foram levantados sete riscos, sumarizados na Tabela 8.

*Tabela 8 - Principais riscos para adoção do EUD*

<b>Autores</b>	<b>Características</b>	<b>Considerações</b>
<b>Cale (1994)</b>	Baixa qualidade das Aplicações Desenvolvidas	A UDA pode ter uma baixa qualidade uma vez que seu desenvolvimento é rápido e normalmente não há preocupações com os aspectos de qualidade da aplicação, como exemplo, o seu código gerado.
	Perda do controle do desenvolvimento da aplicação	Se não monitorado, o desenvolvimento pode gerar prejuízos para a organização, como por exemplo, a não utilização da aplicação desenvolvida.
<b>Alavi &amp; Weiss (1985-</b>	Utilização ineficaz dos	Utilizar tempo, pessoas,

<b>1986); Davis e Srinivasan (1988); O'Donnell (1987)</b>	recursos	recursos e não obter um produto final.
<b>Alavi e Weiss, (1985; 1986); Brancheau e Brown (1993)</b>	Ameaça à segurança dos dados	Não há preocupação com a segurança quando o usuário final está desenvolvendo. Designar um especialista pode ser uma solução viável em alguns casos.
<b>Alavi &amp; Weiss (1985; 1986)</b>	Ameaça à integridade dos dados	
<b>Alavi &amp; Weiss (1985; 1986); Brancheau &amp; Brown (1993)</b>	Resolver o problema errado	Resolve um problema temporário ou com baixa relevância.
<b>McGill (2004)</b>	Abandono da UDA pelos usuários utilizadores	A organização, ao perder seu funcionário usuário final, pode perder o conhecimento daquele usuário desenvolvedor, e com isso a UDA pode ser descontinuada ou perdida.

## 2.8 Tendências em EUD e EuC

O EUD tem se tornado uma prática comum no desenvolvimento de software. Scaffidi et al. (2005) estimaram que no ano de 2005 o número de usuários finais desenvolvedores seria de aproximadamente 55 milhões e de programadores profissionais seria de 3 milhões nos Estados Unidos. Uma nova estimativa foi realizada pelos autores e foi projetado um aumento de aproximadamente 63% do número de usuários finais para o ano de 2012, totalizando aproximadamente 90 milhões de usuários finais, incluindo utilizadores de planilhas e bancos de dados e pessoas que se autodeclararam programadores.

Além do aumento de usuários finais, eles estão criando cada vez mais aplicações com significado tanto para si quanto para organização em que trabalham, além disso, comunidades online emergentes no âmbito do EUD estão crescendo constantemente (STOLLE; ELBAUM; SARMA, 2011). Estas comunidades facilitam o compartilhamento de conhecimento e a

reutilização de código, como por exemplo, os repositórios públicos de mashups no Yahoo! Pipes (STOLLE; ELBAUM; SARMA, 2012), animações no Scratch (SCRATCH, 2016) e scripts de modificação de páginas da web em Userscripts (USERSCRIPTS, 2016).

O EUD tem compreendido diferente campos de pesquisa como a Interação Humano Computador (HCI), a Engenharia de Software (SE), o Trabalho Corporativo Apoiado por Computadores (CSCW) e a Inteligência Artificial (AI). Além disso, existem conceitos inerentes ao EUD que já são alvos de estudos como o *tailorability*, a configurabilidade, a programação pelo usuário final, a usabilidade, a programação visual, a programação natural e a programação por exemplo Lieberman et. al (2006).

O EUD também se centra nas atividades de desenvolvimento ou modificação do código a partir de uma não especialista de software. Sobretudo, há alguns focos de pesquisas que trabalham na abstração do design das linguagens (FISCHER, 1994; REPENNING, 1993), modos de interfaces de comunicação (BURNETT et al., 1995; GREEN e PETRE, 1996), abordagens de Inteligência Artificial (IA) (LIEBERMAN, 2001) e fatores humanos do EUD (NARDI e MILLER, 1991). Tais pesquisas, buscam o aprimoramento das tecnologias utilizadas bem como a relação do usuário não-especialista com o código (MEHANDIJEV, SUTCLIFFE, LEE, 2006).

É possível identificar algumas dificuldades nas pesquisas da área. Mehandjiev, Sutcliffe e Lee (2006) relatam dificuldades em encontrar pesquisas que exploram problemas e conflitos a partir da formalização do EUD. Em contrapartida, é mais comum encontrar relatos de descobertas os quais não fornecem conhecimento necessário para decisões políticas de implantação do EUD.

Ko et al. (2011) apontam que alguns trabalhos recentes sobre o EUD estão se movendo em várias direções ao mesmo tempo: para mais plataformas e paradigmas, incluindo a web e dispositivos móveis; para explorar boas práticas da engenharia de software, incluindo especificação e reutilização de código; para um foco mais amplo em domínios de aplicação, ao invés de paradigma de linguagem único. Além disso, os autores afirmam que a Web tem se tornado uma plataforma dominante para a aprendizagem da programação pelos usuários finais, bem como, para apoiar suas atividades de desenvolvimento do EUD.

### 3 ESTRATÉGIA DE ADOÇÃO DE DESENVOLVIMENTO

A demanda por soluções de software, de planilhas a sistemas, para a área de Tecnologia da Informação (TI) das organizações, públicas ou privadas, cresce constantemente. Comumente, as organizações adotam a estratégia de possuírem uma área de TI centralizada, responsável por suprir as demandas de TI das áreas de negócio, sejam públicas ou privadas. Outra estratégia que vem sendo adotada é descentralização da TI, com a replicação dos serviços de TI nas áreas de negócio.

Paralelo a esse movimento de descentralização, surge o desenvolvimento de soluções de software pelo usuário final, conhecido por End-User Development (EUD) (MCLEAN, 1979; LIEBERMAN et al., 2006; BARKER, 2007; MACÍAS, 2012).

O EUD tem se tornado uma prática, seja pelo usuário não profissional de desenvolvimento de software que busca desenvolver soluções de software para uso próprio (STOLEE; ELBAUM; SARMA, 2011), ou por organizações que empregam uma estratégia para fornecer às áreas de negócio, recursos como profissionais, processos, ferramentas, arquitetura, entre outros, para que essas áreas de negócio desenvolvam seu software (MEHANDIJEV; SUTCLIFFE; LEE, 2006). Contudo, muitos representantes das áreas de negócio podem vir a desenvolver soluções de software sem conhecimento e controle pela área de TI da organização.

Os autores Mehandjiev, Sutcliffe e Lee (2006) definem EUD como um paradigma sócio técnico focado em permitir que usuários finais, normalmente não profissionais de desenvolvimento, de soluções de softwares (sistemas interativos), se engajem na modificação, extensão e até criação de artefatos. Em 2006, os autores realizaram um levantamento de percepções da adoção de EUD em um contexto organizacional, envolvendo pesquisadores de universidades, profissionais da indústria e do governo. O objetivo foi identificar percepções da adoção e utilização do EUD, bem como benefícios e motivações, o grau de uso e a compreensão de tecnologias e ferramentas EUD.

O resultado da pesquisa de Mehandjiev, Sutcliffe e Lee (2006) gerou uma abordagem de adoção e gestão do EUD a partir do processo de análise resultante da identificação de oito temas emergentes e três conceitos básicos.

Os temas compreenderam: estratégia da TI; demanda organizacional para EUD; custo-benefício para os usuários finais; ferramentas adequadas às tarefas do EUD; gestão e controle; metodologias EUD; reconhecimento e responsabilização do trabalhador; e profissionalização. E os conceitos compreenderam: motivação para EUD; riscos decorrentes de práticas EUD; e controle das práticas EUD.

Os *temas emergentes* fornecem uma visão geral das questões apresentadas em publicações e discussões de pesquisadores em EUD, e *os conceitos básicos* surgiram a partir

do levantamento da demanda pela adoção do EUD, focados nos aspectos práticos da abordagem.

Mehandjiev, Sutcliffe e Lee (2006) ressaltam que o tema emergente *estratégia de TI* abrange alguns aspectos técnicos e estratégicos, e afirmam que a adoção do EUD pode funcionar como uma estratégia de ação rápida as modificações nas operações de negócio pelos usuários finais. McGill (2004) acredita que o EUD pode evitar a burocracia existente no desenvolvimento de software centralizado, podendo funcionar como um motivador para o usuário final. Sobretudo, as tecnologias podem contribuir para essa rápida capacidade modificação, uma vez que reduz o custo de aprendizagem de programação para o usuário final (SHAYO; GUTHRIE; IGBARIA, 1999). Fischer et al. (2004) ressalta que a área de TI deve adotar estratégias que permitam usuários finais compartilhar conhecimento entre si.

Quanto à demanda organizacional, o EUD pode demandar um grande esforço da TI para monitorar e controlar as aplicações desenvolvidas e a capacitação dos usuários finais. No entanto, McGill (2004) afirma que é possível aumentar a capacidade de atendimento das demandas TI para o negócio a partir do EUD, pois permite a TI focar no desenvolvimento de software complexo e realocar software menos prioritários para a área de negócio. Sutcliffe (2005) reconhece a necessidade de quantificar os benefícios oriundos do EUD, a fim de motivar usuários finais a desenvolver suas próprias aplicações com apoio da TI.

O tema emergente *custo-benefício* ressalta a importância de balancear os custos inerentes à adoção e manutenção do EUD com os benefícios provindos pelo EUD (SUTCLIFFE, 2005). Aghae e Pautasso (2013) apresentam que um dos principais custos envolvidos no EUD é a aprendizagem de programação que os usuários finais deverão realizar. Para minimizar este custo, os autores sugerem a utilização de ferramentas que facilitem os usuários finais a desenvolver suas aplicações, assim será possível, ao final do desenvolvimento, usuários finais obterem uma solução compatível com suas necessidades. Mehandjiev, Sutcliffe e Lee (2006) ressaltam que a combinação de custo benefício fornecer uma motivação para que os usuários finais utilizem do EUD para soluções rápidas em suas organizações.

Quanto às *ferramentas* adequadas para o EUD, Mehandjiev, Sutcliffe e Lee (2006) ressaltam que a utilização de ferramentas adequadas para o contexto do negócio pode reduzir a curva de aprendizagem e a complexidade do desenvolvimento pelos usuários finais. As ferramentas têm um papel fundamental no desenvolvimento, pois os usuários finais são, comumente, pessoas não especialistas em desenvolvimento de software. Logo, fornecer ferramentas que permitam usuários participarem de forma ativo no desenvolvimento de soluções de software é essencial para o EUD (FISCHER et al., 2004).

O tema emergente *gestão e controle* refere-se às atividades de gestão e controle da TI em relação ao EUD. Mehandjiev, Sutcliffe e Lee (2006) afirmam uma preocupação que a TI deve ter sobre o EUD é a qualidade das aplicações (software) desenvolvidas. Eles sugerem

que a TI deve ter controle sobre o grau de qualidade que as aplicações devem ter e também sugerem a utilização de políticas que definam quais aplicações devem ser desenvolvidas pelo EUD ou pelo desenvolvimento centralizado na TI.

Quanto às *metodologias* no EUD, Jennex (2005) afirma que a utilização de metodologias tradicionais de desenvolvimento de software pode sobrecarregar usuários finais. Por outro lado, Fischer e Giaccardi (2004) propõem a utilização de abordagens voltadas para o *meta design* focando no desenvolvimento colaborativo em um conjunto de usuários finais, mas sem a utilização de políticas rígidas os quais possam engessar o desenvolvimento de soluções de software por usuários finais.

O tema emergente *reconhecimento e responsabilização* do trabalhador foca nas formas de reconhecer e estimular a responsabilidade do usuário final para o desenvolvimento de software. Usuários finais, ao se disporem a realizar as atividades de desenvolvimento de software, assumem novas responsabilidades que normalmente não de sua competência. Neste sentido, Mehandjiev, Sutcliffe e Lee (2006) ressaltam a importância de reconhecer essa responsabilidade extra que os usuários finais assumem, fornecendo suporte adequado para que eles possam tomar as decisões acertadas durante o desenvolvimento. Os autores sugerem que os usuários finais, ao cumprirem com suas metas, sejam gratificados, aumentam o seu entusiasmo e a sua motivação. Sutcliffe (2005) confirma que quanto maior for o comprometimento dos usuários finais e necessidade da aplicação, mais motivados os usuários participam do desenvolvimento.

Quanto à *profissionalização*, Mehandjiev, Sutcliffe e Lee (2006) afirmam que são práticas e métodos de teste que podem ser percebidos como importante para o desenvolvimento pelo usuário final, isto é, maneira de melhorar a qualidade das aplicações originadas a partir do EUD por meio de utilização de testes.

A partir dos temas emergentes identificados e dos conceitos básicos compreendidos, Mehandjiev, Sutcliffe e Lee (2006) apresentam em uma abordagem sócio técnica para adoção e gestão do EUD, a Figura 2 compreende este modelo. No centro do modelo há três núcleos de atividades apresentadas como objetos de atenção, são eles: a Análise de Adoção do EUD; a Implantação do EUD; e a Gestão do EUD. São incorporados facilitadores ao modelo para auxiliar os três núcleos principais, o objetivo dos facilitadores é fornecer base para execução bem-sucedida. Esses facilitadores são apresentados no formato de elipse no modelo.

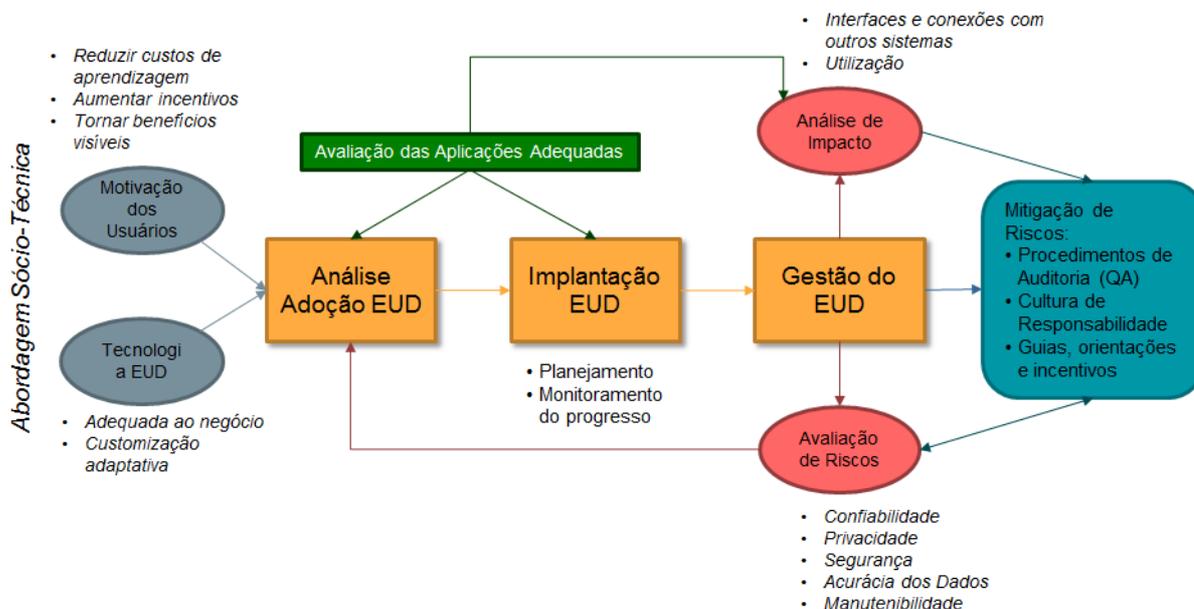


Figura 2 - Abordagem Sócio Técnica na Gestão do EUD. Fonte: Mehandjiev, Sutcliffe e Lee (2006)

O centro do modelo compreende processo de gestão onde o núcleo do controle do EUD é estabelecido. Os facilitadores apresentados na seção anterior servem de insumo para o núcleo de gestão. As próximas seções exploram os três núcleos de atividades do processo de gestão e sua relação com os quatro facilitadores baseados nos temas emergentes.

### 3.1 Tecnologias e Ferramentas EUD

Mehandjiev, Sutcliffe e Lee (2006) afirmam que tecnologias para o desenvolvimento de aplicações para o usuário final é difícil de ser aprendida atualmente. No entanto, há esforços em tentar permitir que ferramentas facilitem a programação pelo usuário final. Um estudo da arte em EUSE (KO et al., 2011) revela que muitas tecnologias emergentes como *design* de processos, reutilização e compartilhamento de código, exploradores de erros entre outros estão surgindo para facilitar a vida dos usuários finais desenvolvedores. Outro ramo de pesquisa que busca facilitar o desenvolvimento pelo EUD é a área de IHC. Alguns estudos como (TEICHE, 2009) estão possibilitando os usuários finais desenvolver, com facilidade, aplicações que atendam as expectativas de negócio.

Segundo Mehandjiev, Sutcliffe e Lee (2006), os usuários finais, mesmo com dificuldades em desenvolver aplicações com apoio ferramental, devem ser motivados. Assim, os usuários finais investem o esforço necessário para o desenvolvimento de aplicações e pode compartilhar com outros usuários desenvolvedores suas dificuldades e superações (FISCHER et al., 2004).

Com o auxílio da tecnologia, é possível criar comunidades de apoio que permitam interação entre TI e usuários finais. Fischer e Giaccardi (2004) sugerem a utilização de um *framework* que permite usuários finais e membros da equipe de TI colaborarem em um ambiente único e integrado. Esta abordagem, chamada pelos autores de *meta-design*, possibilita que a TI e a área de negócio desenvolvam juntas aplicações que atendam as necessidades do negócio. Pipek e Kahler (2006) sugerem a utilização de ferramentas que apoiem os usuários finais a compartilhar desenhos, códigos reutilizáveis, entre outros exemplos.

Mehandjiev, Sutcliffe e Lee (2006) sugerem a utilização de ferramentas que se adequem ao negócio e que possibilitem a customização de acordo com a necessidade de negócio. Essas ferramentas podem incluir um grau de dificuldade que usuários finais não podem superar sozinho, por isso, é necessária a construção de treinamentos adequados para utilização das ferramentas. Com isso, usuários finais poderão ter uma formação mínima para o desenvolvimento de aplicações a partir do EUD.

### 3.2 Motivação dos Usuários

Usuários finais não programadores podem ter dificuldade tanto para aprender a programar quanto para programar durante um determinado período de tempo (PANTAZOS; LAUESEN, 2016). Contudo, é essencial que os usuários sejam motivados a desenvolver suas próprias aplicações com incentivos e funções cuidadosamente projetadas (MEHANDIJEV; SUTCLIFFE; LEE, 2006). Algumas maneiras de incentivar os usuários são:

- Utilizar comunidades de apoio aos usuários finais com o objetivo da TI dar suporte para melhorar a experiência durante o desenvolvimento, bem como reduzir a lacuna existente entre o usuário final e a atividade de desenvolvimento (NARDI, 1993);
- Tornar os benefícios e histórias de sucesso visíveis aos usuários finais da organização (MEHANDIJEV; SUTCLIFFE; LEE, 2006);
- Compartilhar o conhecimento e soluções de outros usuários finais para novos (FISCHER, 2010);
- Envolver os usuários finais no desenvolvimento para que eles possam ter a sensação de que são importantes para o desenvolvimento (MEHANDIJEV; SUTCLIFFE; LEE, 2006);
- Recompensar o sucesso do desenvolvimento de aplicações e envolvimento do usuário final (SUTCLIFFE, 2005).

Além dos incentivos, Siricharoen (2012) ressalta a importância de conhecer o perfil do usuário final para definir sua função e grupos de atividades no desenvolvimento de uma solução de software.

Carter e Henderson (1990) propõem à redução do custo de aprendizagem dos usuários como um benefício o qual motivam eles a continuar desenvolvendo aplicações. Para isso, é necessário tempo para que os usuários aprendam novas ferramentas e obtenham treinamento adequado. Este tempo permite ao EU errar e aprender com seus erros.

Outros benefícios oriundos da adoção do EUD os quais podem motivar os usuários finais e a alta administração de uma organização são:

- Melhora a tomada de decisão do usuário final durante o desenvolvimento, uma vez que ele participe do desenvolvimento suas decisões são mais precisas e objetivas (BRANCHEAU; BROWN, 1993);
- Redireciona demandas de baixa prioridade de desenvolvimento na TI para a área de negócio (SHAYO; GUTHRIE; IGBARIA, 1999);
- Reduz a sobrecarga da TI e aumenta a capacidade de entrega permitindo que ela se preocupe apenas com software com maior prioridade (MCGILL, 2004);
- Melhora a relação da TI com os usuários finais (MCGILL, 2004);
- Melhora a satisfação geral do usuário quanto a TI e a aplicação desenvolvida por ele (DOLL; TORKZADEH, 1989; LIN; SHAO, 2000);
- Melhora o desempenho do usuário final utilizador da aplicação, caso ele a tenha desenvolvido (AMOROSO, 1988).

### **3.3 Análise de Adoção do EUD e Avaliações de Aplicações Adequadas**

Adotar o EUD em uma organização pode ser uma tarefa complexa. Pipek e Kahler (2006) sugerem que organizações que desejam adotar o EUD como uma estratégia de desenvolvimento de software devem realizar uma avaliação de sua estrutura. Segundo eles, a TI da organização deve avaliar se o pessoal da área de negócio tem maturidade para desenvolver aplicações software com apoio da TI e se sua infraestrutura tecnológica permite o desenvolvimento pelo usuário final. Sutcliffe (2005) ressalta que ferramentas que permitem, usuários não-especialistas em desenvolvimento de software, desenvolver e implantar aplicações podem ser necessárias em uma organização que deseja adotar o EUD.

Além da avaliação para adoção do EUD, é necessário que a organização elabore políticas as quais definam as aplicações que deve ser desenvolvida pelo EUD ou pelo desenvolvimento centralizado na TI (MEHANDIJEV; SUTCLIFFE; LEE, 2006).

Alguns aspectos devem ser levados em consideração na hora de avaliar se uma aplicação é adequada para o desenvolvimento pelo EUD: domínio da aplicação (COSTABILE; PICCINNO, 2003); perfil de usuário final (COTTERMAN; KUMAR, 1989); segurança e integridade dos dados (BRANCHEAU; BROWN, 1993); relevância da aplicação (MCGILL, 2004); e riscos relacionados à qualidade da aplicação (ALAVI; WEISS, 2015).

### 3.4 Implantação e Gestão do EUD

Planejar a execução do desenvolvimento pelo EUD é importante. Mehandjiev, Sutcliffe e Lee (2006) sugerem que definir processos e métodos é uma alternativa viável para auxiliar o usuário final a não se perder ao longo de desenvolvimento.

No entanto, deve-se evitar metodologias tradicionais de desenvolvimento, as quais podem trazer pouca flexibilidade para as atividades a serem realizadas pelos usuários. Os usuários finais precisam ser capazes de realizar as atividades de desenvolvimento de modo padronizado, mas também devem ter liberdade para interagir com outras atividades além das atividades de desenvolvimento, e interagir com outros usuários finais e com a TI (JENNEX, 2005).

Fischer (2010) ressalta a importância da cultura da participação no EUD. Os usuários finais devem ser participantes ativos no processo desenvolvimento e potenciais colaboradores de novos usuários finais, reforçando a colaboração mútua entre antigos e novos usuários finais.

Durante o desenvolvimento, Mehandjiev, Sutcliffe e Lee (2006) sugerem que é importante monitorar o progresso do usuário final. Para isto, a área de TI pode:

- Apoiar o desenvolvimento fornecendo profissionais colaborados, os quais são alocados para ajudar em possíveis dúvidas que surjam durante o desenvolvimento (FISCHER; GIACCARDI, 2004);
- Criar comunidades colaborativas contendo antigos e novos usuários finais e profissionais de TI para gerar discussões sobre aprendizagem, qualidade, erros e compartilhamento de experiências (NARDI, 1993; FISCHER et al., 2004);
- Criar treinamento das tecnologias EUD a serem utilizados pelos usuários finais disponibilizando-as em tempo integral para os usuários finais desenvolvedores (JENNEX, 2005; BURNETT, 2009);
- Motivar usuários finais, por meio de políticas compartilhamento ligadas à gestão de conhecimento da organização, a compartilhar história de sucesso e fracasso no desenvolvimento EUD (MEHANDIJEV; SUTCLIFFE; LEE, 2006).

Para Mehandjiev, Sutcliffe e Lee (2006), o controle das práticas do EUD pode ser baseia em duas principais atividades: Análise de Impacto e Avaliação de Risco. As duas seções a seguir apresentam detalhes sobre essas atividades.

### 3.5 Análise de Impacto

No decorrer do desenvolvimento de uma aplicação, alguns aspectos podem impactar diretamente a aplicação ou usuário final como, por exemplo, interface de comunicação com

outros sistemas (MEHANDIJEV; SUTCLIFFE; LEE, 2006) e tomada de decisão pelo usuário (BRANCHEAU; BROWN, 1993).

Interfaces de comunicação e conexões com outros sistemas podem ser simples de implementar a depender do domínio da aplicação. Alguns sistemas podem permitir somente a leitura de dados em sua base de dados como também pode ser necessário à criação de uma nova base de dados mapeada com a base original. Essa decisão pode ser tomada mediante a avaliação de aplicações adequada, por meio de diretrizes claras e bem definidas (JENNEX, 2005).

Os usuários finais devem ter liberdade para tomar decisões mediante ao desenvolvimento de aplicações com base no apoio fornecido da TI. No entanto, a TI deve monitorar essas decisões a fim de evitar experiências não benéficas para os usuários (SUTCLIFFE, 2005). Segundo Shayo et al. (1999), o EUD pode melhorar a tomada de decisão dos usuários de acordo com a evolução do projeto e com novos projetos.

Mehandjiev, Sutcliffe e Lee (2006) afirmam que a análise de impacto ajuda no equilíbrio entre velocidade de desenvolvimento e qualidade do produto gerado. Neste sentido, é importante definir uma forma de monitoramento do desenvolvimento e da qualidade. Alavi e Weiss (2015) sugerem a utilização de auditorias para garantir a qualidade da aplicação após o desenvolvimento do software.

Mehandjiev, Sutcliffe e Lee (2006) ainda sugerem analisar impactos e riscos por tipos de sistemas, como exemplo, sistemas que acessam informações na base principal de dados da organização, e sistemas com bases de dados próprias como aplicação de agendamento de visitas. Essa estratégia pode diminuir o esforço da TI para monitorar o desenvolvimento pelo EUD.

### **3.6 Avaliação e Mitigação de Riscos**

Mehandjiev, Sutcliffe e Lee (2006) ressaltam que alguns riscos devem ser avaliados e gerenciados, bem como devem ser insumos para avaliação de aplicações adequadas para EUD. Eles elencam os riscos para EUD nas áreas da confiabilidade, facilidade de manutenção, segurança, precisão e privacidade dos dados.

Para mitigar os riscos apresentados elencados por (MEHANDIJEV; SUTCLIFFE; LEE, 2006), algumas abordagens podem ser viáveis como: estabelecimento de diretrizes para acesso a dados (JENNEX, 2005); avaliação de qualidade e auditorias (ALAVI; WEISS, 2015); estimular a cultura da responsabilidade (MEHANDIJEV; SUTCLIFFE; LEE, 2006); e utilização de bases de conhecimento de projetos EUD (MCGILL, 2004).

O estabelecimento de políticas e diretrizes é uma solução recorrente para controle das atividades como visto nos tópicos anteriores. Para o acesso de dados, é comum que organizações utilizem de políticas institucionais, legislação, protocolos de acessos entre

outros para restringir acesso aos dados. Para o EUD, essas diretrizes podem ser utilizadas de acordo com o domínio da aplicação. Aplicações com escopo corporativo podem ser desenvolvidas pelo desenvolvimento centralizado na TI, enquanto que aplicações departamentais podem ser desenvolvidas pelo EUD (MCGILL, 2004).

A avaliação de qualidade e utilização de auditorias também é uma solução recorrente no que tange a qualidade. Elas permitem que a TI monitore o desenvolvimento e evite que erros possam ser cometidos novamente em outros projetos EUD (CALE, 1994; ALAVI; WEISS, 2015).

Mehandjiev, Sutcliffe e Lee (2006) sugerem a estimular a cultura da responsabilidade no EUD mostrando ao usuário final que ele possui responsabilidade sobre o que está sendo desenvolvido. Neste sentido, os usuários finais terão cautela na hora de tomar decisões. No entanto, Mehandjiev, Sutcliffe e Lee (2006) ressaltam que ao mesmo tempo em que o usuário é cobrado, ele também deve ser motivado, equilibrando o desejo de continuar desenvolvendo por meio do EUD.

A utilização de bases de conhecimento pela TI pode aumentar as chances de sucesso do desenvolvimento de uma aplicação EUD (MCGILL, 2004). Isso porque a TI poderá acumular maturidade e experiências, permitindo ela tomar decisões rápidas e eficientes com base nos problemas já compreendidos e registrados em sua base de conhecimento de desenvolvimento pelo EUD. Ou seja, quanto maior for o conhecimento de riscos e impactos conhecidos pelo TI no EUD, maior será a capacidade de suporte que a TI pode fornecer aos usuários finais ao longo do desenvolvimento de uma solução de software.

Mehandjiev, Sutcliffe e Lee (2006) ressaltam que políticas, diretrizes e padrões não devem ser engessados, pois podem limitar a liberdade dos usuários durante o desenvolvimento. Neste sentido, cabe a TI adequar este aspecto dentro da organização.

## 4 ANÁLISE DE ESTRATÉGIAS DE DESENVOLVIMENTO DESCENTRALIZADO EM INSTITUIÇÕES PÚBLICAS

### 4.1 Estudo de Caso A

O Órgão Público A é responsável pela fiscalização, auditoria e julgamento de contas de outros órgãos. Foi o órgão que serviu de exemplo para que os outros também adotassem o desenvolvimento descentralizado.

A TI do órgão conta com 136 servidores, 79 estagiários e 24 terceiros prestadores de serviços.

#### 4.1.1 Histórico do EUD

Em 2007, a área de TI do Órgão Público A possuía aproximadamente 200 colaboradores sendo 100 servidores, 60 estagiários e 40 terceirizados. Nos anos posteriores, somaram-se à equipe de TI dezenas de usuários, gestores e clientes, que participam diretamente do provimento das soluções de TI que suportam e alavancam processos de trabalho do órgão.

A Secretaria A era responsável direta pelo provimento de soluções de tecnologia da informação, bem como pela administração dos ativos de TI. Existe no Órgão Público A um comitê de governança atuante desde 2001.

Com a popularização das planilhas eletrônicas, os usuários finais do órgão criam soluções que informatizam suas atividades para resolver problemas específicos. Neste sentido, sempre houve usuários desenvolvedores no órgão, que resolviam problemas locais com ferramentas de escritório ou até mesmo com linguagens de programação. A área de TI não participava ou não tinha conhecimento deste desenvolvimento e isto decorria da dificuldade em atender as demandas pontuais dos clientes da TI.

No biênio 2007/2008, o Órgão Público A reconheceu que o desenvolvimento departamental era uma realidade no órgão e decidiu adotar uma estratégia para melhor lidar com esse cenário. Uma iniciativa para “implantar ferramenta para desenvolvimento de sistemas departamentais e prover suporte aos usuários na utilização da referida ferramenta” foi incluída no Plano Diretor de Tecnologia da Informação de 2007/2008, iniciando-se uma nova era de relacionamento entre TI e seus clientes. Nesta ocasião foi também instituído na estrutura organizacional de Tecnologia da Informação unidade específica de governança e apoio ao desenvolvimento descentralizado, visando garantir a qualidade, segurança e padronização das aplicações desenvolvidas pelos diversos departamentos.

Em 2009, foram realizadas algumas modificações na estrutura administrativa do órgão. A nova presidência havia definido um projeto que exigia dedicação exclusiva de todos os colaboradores da TI. Para cumprir as metas que foram estabelecidas, recursos foram alocados

à Secretaria A, entre eles, a reestruturação da área, dividindo-a em duas secretarias: uma de infraestrutura e outra de desenvolvimento de soluções, e a criação de uma assessoria.

#### 4.1.2 Estratégia de adoção do EUD: áreas do negócio, critérios para seleção das soluções, suporte dado pela TI

Ao longo dos anos, a unidade específica de governança e apoio ao desenvolvimento descentralizado ganhou maior atenção e passou a ser responsável por selecionar estagiários da área de TI para o desenvolvimento das aplicações e por realizar um acompanhamento do desenvolvimento dessas aplicações. Em 2013, este serviço contava com sete servidores e acompanhava 62 projetos que correspondiam a 123 sistemas.

Atualmente a unidade específica de governança e apoio ao desenvolvimento descentralizado é responsável por apoiar o desenvolvimento descentralizado, tendo como atividades: governar o modelo de desenvolvimento descentralizado; definir padrões; verificar as conformidades; apoiar a seleção de desenvolvedores; treinar e capacitar os servidores para o desenvolvimento de soluções; prestar consultoria e suporte para os desenvolvedores; fomentar a comunidade de desenvolvedores; acessar os dados corporativos; e gerir o ambiente e infraestrutura para desenvolvimento.

A estratégia adotada pelo órgão para determinar o destino das demandas por soluções de software (centralizado ou descentralizado) é baseado no perfil da solução. Neste sentido, soluções de TI departamentais são desenvolvidas pela estratégia de desenvolvimento descentralizada e as soluções de TI corporativas são desenvolvidas pela TI central. No entanto, o desenvolvimento descentralizado de demandas corporativas requer aprovação do Comitê de TI. Na Figura 3 apresenta-se esta relação.

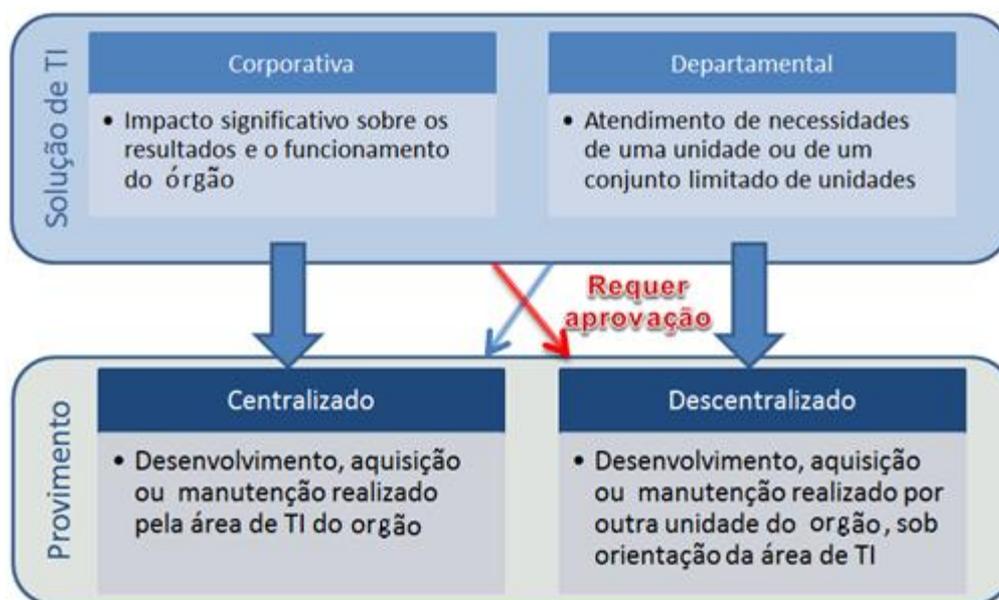


Figura 3 - Estratégias de desenvolvimento de soluções de TI do Órgão Público A

Para a execução dos projetos EUD, cada uma das áreas de negócio deve selecionar e alocar um servidor (normalmente é nomeado alguém que possua alguma experiência prévia na área de TI) para o desenvolvimento descentralizado. Então, a demanda por EUD deve ser cadastrada em um sistema específico, a partir do qual vai ocorrer o acompanhamento pelo serviço de apoio ao desenvolvimento descentralizado.

#### 4.1.3 Tecnologias Adotadas

Durante alguns anos, os servidores do Órgão Público A utilizaram diversas tecnologias para o desenvolvimento informal nas áreas, tais como Microsoft Access, PHP, ASP, JSP entre outros. Atualmente a ferramenta utilizada pelo órgão para a implementação do desenvolvimento descentralizado é o *Oracle Application Express* ou simplesmente Apex.

A escolha pelo Apex se deu devido ao fato de o órgão possuir ferramentas e licenças Oracle (ou seja, a aquisição do Apex não resultou em custos adicionais de licenciamento). Além disso, com a utilização do Apex, a integração com as ferramentas já utilizadas (da Oracle), tornava-se simples.

Cada aplicação desenvolvida pelo desenvolvimento descentralizado possui sua própria base de dados, realizando a comunicação com a base central (somente leitura) a partir de *views*, conforme apresentado na Figura 4.

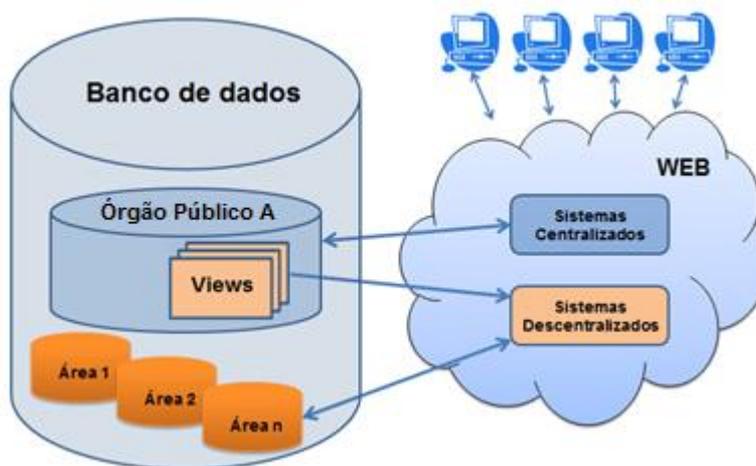


Figura 4 - Visão do consumo de dados dos sistemas centralizados e descentralizados do Órgão Público A

#### 4.1.4 Qualidade e Segurança

Durante a formalização do desenvolvimento descentralizado no Órgão Público A, foi desenvolvido o PDESC (Processo de Desenvolvimento Descentralizado). O PDESC fornece

um detalhamento do ciclo de vida, atividades, papéis, padrões de interfaces, usabilidade, documentação entre outras informações necessárias para a realização do desenvolvimento descentralizado dentro do órgão. No entanto, hoje o modelo não é mais atualizado, sendo substituído por treinamentos e capacitações online, disponibilizadas no *website* do órgão.

Atualmente não há um processo de garantia da qualidade. Cada secretaria é responsável pelas suas próprias aplicações. Sobretudo, algumas medidas são executadas a fim de mitigar riscos, como a limitação do consumo dos dados da base da Oracle, evitado interferências das aplicações departamentais em corporativas de uso sensível para a instituição.

#### **4.1.5 Impactos e Benefícios**

Os principais benefícios observados com a implantação do desenvolvimento descentralizado foram:

- Melhorias na governança e gestão de TI;
- Ampliação da capacidade produtiva da área de TI;
- Ampliação de unidades atendidas e da satisfação dos clientes;
- Possibilidade de desenvolvimento de soluções de natureza departamental ou corporativa de médio e grande porte;
- Elevada produtividade e robustez;
- Redução de custos para o desenvolvimento de aplicações;
- Aumento na segurança de informações.

Algumas limitações encontradas na estratégia de desenvolvimento descentralizado de software pelo usuário final foram:

- Capacidade de atendimento da equipe de TI (média de analista por sistema acima da capacidade);
- Necessidade de contratação de estagiários;
- Ferramenta RAD de desenvolvimento adequada para sistemas transacionais (CRUD);
- Ferramenta RAD de desenvolvimento e padrões definidos impõem restrições de usabilidade;
- Resultados a médio e longo prazos (pouca experiência dos desenvolvedores).

#### **4.1.6 Lições Aprendidas**

Uma das principais lições aprendidas foi a importância de se possuir um processo de trabalho a ser seguido e exigir uma documentação mínima para guiar os usuários finais durante o desenvolvimento de suas soluções. Segundo o Órgão A, é fundamental que haja um controle central sobre o desenvolvimento descentralizado.

Além disso, foi notado pelo órgão que a cultura organizacional é um fator crítico de sucesso para adoção do EUD. Neste sentido, deve-se proporcionar a cooperação entre TI e a área de negócio, a confiança nos usuários finais desenvolvedores e o compartilhamento de responsabilidade entre TI e a área de negócio.

Treinamento e capacitação também foram fatores chaves para o sucesso do EUD, bem como ferramentas adequadas para o desenvolvimento de soluções de TI a partir do EUD e uma equipe de suporte de TI qualificada.

#### 4.1.7 Perspectivas Futuras

O Órgão Público A já vem fazendo uso de técnicas de desenvolvimento descentralizado nos últimos anos, sendo um dos exemplos mais completos do uso da abordagem. Os documentos e diretivas seguidas são atualizados constantemente, e o foco principal é manter o processo. Além disso, foi iniciado um processo de ampliação do EUD para atividades de TI além do desenvolvimento de sistemas (e.g. confecção de painéis de informação).

#### 4.1.8 Aplicações Desenvolvidas no modelo descentralizado/EUD

Após a implementação do desenvolvimento descentralizado no Órgão Público A, mais de 200 sistemas já foram produzidos seguindo este modelo. Atualmente, o Órgão Público A conta com 141 sistemas em operação, 34 em desenvolvimento, 28 em planejamento e 75 sistemas descontinuados. A Tabela 9 apresenta alguns desses sistemas no órgão.

*Tabela 9 – Exemplos de sistemas desenvolvidos no modelo descentralizado/EUD*

Sigla	Nome	Classificação	# de telas	Descrição
<b>Sispar</b>	Sistema de gerenciamento para Aspar	Departamental	Mais de 150 telas	A solução tem por finalidade auxiliar os servidores da Assessoria Parlamentar no acompanhamento de projetos de lei de interesse do Órgão Público A
<b>Sisconjur</b>	Sistema de acompanhamento processual da Consultoria	Departamental	Mais de 100 telas	A solução tem por finalidade viabilizar o registro e o acompanhamento das ações

	Jurídica do Órgão Público A			e processos de interesse do órgão que tramitam no Poder Judiciário em todas as instâncias.
<b>Sigeste</b>	Sistema de gerenciamento de estágio estudantil	Corporativo	Mais de 100 telas	Sistema de gerenciamento de estágio estudantil do Órgão Público A

## 4.2 Estudo de Caso B

O Órgão Público B é uma entidade que compõe o Poder Administrativo Federal.

O desenvolvimento por usuário final no Órgão Público B iniciou em 2012 a partir da experiência do Órgão Público A. A principal motivação para a adoção do modelo naquele momento foi prover uma ferramenta aos usuários, alternativa ao uso de planilhas, para resolver as demandas das áreas de negócio.

Inicialmente, havia três modalidades de desenvolvimento: desenvolvimento realizado por servidor do Órgão Público B, desenvolvimento realizado por estagiário alocado nas áreas de negócio e desenvolvimento por estagiário na TI. O grupo de usuários-desenvolvedores era composto por dois servidores da área de negócio e um estagiário de TI. Atualmente o grupo conta com nove servidores do negócio dedicados em projetos de sistemas EUD e cinco servidores de TI alocados em áreas de trabalho distintas, contudo não há mais estagiários alocados nas áreas de negócio, apenas dentro da TI.

Desde 2012, 38 sistemas foram desenvolvidos neste modelo e encontram-se em manutenção. O Órgão não formalizou o desenvolvimento descentralizado. Sobretudo, o Plano Diretor de TI (PDTI) detalha iniciativas em Apex, ferramenta comum no desenvolvimento descentralizado, mas não descreve um processo de desenvolvimento para este modelo.

### 4.2.1 Estratégia de adoção do EUD: áreas do negócio, critérios para seleção de soluções, suporte dado pela TI

Por regra, todas as demandas de desenvolvimento de software do Órgão Público B são captadas e analisadas de forma centralizada por equipes de análise técnica. A partir desse processo a demanda pode seguir por três caminhos: ser desenvolvida de forma centralizada pela TI tradicional, ser desenvolvida por servidores da área de negócio na ferramenta APEX, ou ainda ser desenvolvida por uma área específica da TI, por um servidor ou estagiário, também em Oracle/APEX.

É possível, no entanto, que alguns desenvolvimentos sejam iniciados na área de negócio, sem passar pelo processo de captação e análise centralizada. Nestes casos, a TI tomará conhecimento do sistema no momento em que for solicitada sua implantação em produção, uma vez que os usuários-desenvolvedores não possuem permissão para este procedimento.

Os critérios para definir se uma demanda será desenvolvida no modelo EUD ou na TI centralizada levam em consideração fatores como a expectativa de tempo para atendimento da demanda, a complexidade da solução, a plataforma em que será utilizada a solução e a possibilidade da necessidade ser atendida por um software de mercado. Em resumo, é avaliado se a necessidade do usuário demandante será plenamente atendida com um sistema Apex, já considerando sua expectativa final da solução.

No início da metodologia de desenvolvimento descentralizado no Órgão Público B, as demandas eram essencialmente departamentais de casos simples para sistematização um controle geralmente feito em planilhas. Com o tempo, as necessidades foram ficando mais complexas, necessitando de um acompanhamento mais dedicado dos servidores. Atualmente, a tecnologia utilizada nesta metodologia, o Apex, tornou-se uma opção de tecnologia de sistemas assim como outras do desenvolvimento centralizado como Java e PHP.

O Órgão hoje conta com 14 áreas de negócio sendo atendidas por equipes centralizadas de TI e, fora da equipe, há cinco servidores que são responsáveis por sistemas, um em cada área, totalizando 19 áreas de negócio envolvidas no desenvolvimento do Órgão Público B. Os usuários-desenvolvedores das áreas de negócio não tem necessariamente formação em TI, mas são usuários curiosos, que veem uma oportunidade de melhorar as suas atividades do dia-a-dia.

Há em torno de oito estagiários que desenvolvem em APEX, eles estão alocados em uma área específica da TI. Não há um processo seletivo específico para a contratação deles.

Grande parte das demandas que são desenvolvidas em APEX são aquelas caracterizadas por funcionalidades de criação, alteração e exclusão de registros.

#### **4.2.2 Tecnologias Adotadas**

Com a experiência do Órgão Público A, o órgão B vislumbrou na gratuidade da licença da ferramenta Oracle/APEX uma vantagem. De acordo com a área responsável pelo desenvolvimento APEX no Órgão Público B, a ferramenta proporciona uma curva rápida de aprendizado e segurança.

Antes da adoção desta tecnologia, os servidores da área de negócio usavam planilhas com grandes volumes de dados (planilhas de 250 colunas) para realizar suas atividades. Os sistemas desenvolvidos em APEX permitiram importar esses dados das planilhas e prover mais segurança e maior produtividade no trabalho, já que, ao contrário das planilhas, vários usuários poderiam acessar e manter os dados de forma concorrente.

Para que estagiários da TI ou servidores passem a utilizar o APEX, eles passam por um treinamento de duas semanas, realizado pela TI. Ao final desta capacitação, eles estão aptos a construir sistemas que sejam capazes de realizar as operações básicas de inclusão, exclusão e alteração de registros em tabelas do banco de dados.

Além do APEX, o Órgão Público B permite desenvolver em outras tecnologias. No entanto, é necessário que a área interessada obtenha uma autorização, a qual poderá ser emitida após a avaliação da tecnologia pela TI.

Quanto à organização de ambientes, a TI separa-os entre ambiente de desenvolvimento, homologação e produção.

#### **4.2.3 Qualidade e Segurança**

O Órgão Público B possui um catálogo de projetos para identificar sistemas já desenvolvidos. Exemplo: sistema de eventos (desde uma reunião com gastos até coisas maiores) tornou-se o SGE. Nesse caso, a TI foi mediadora para que as áreas conversassem e fosse criado um projeto em conjunto.

A TI mantém um portal com um processo para orientar os desenvolvedores. Especificamente para as atividades de requisitos de software, no Órgão Público B a elicitação para as aplicações EUD tem sido realizado pela TI. Essa medida, segundo o órgão, facilita o controle do que tem sido desenvolvido.

O Órgão Público B considera o APEX um ambiente que proporciona segurança. Na versão cinco da ferramenta há bloqueio a alguns tipos de ataque que *derrubavam* o APEX na versão anterior como *SQL injection*.

Há sistemas do Órgão Público B que estão disponíveis para usuários externos como o Sistema A, o Sistema de Transparência, e o Sistema para Localização de Unidades. No caso do Sistema A há um questionário longo que é distribuído em vários estados, que processa um volume considerável de múltiplos acessos.

#### **4.2.4 Impactos e Benefícios**

Um dos benefícios do modelo EUD é a velocidade do desenvolvimento. Segundo o órgão é possível ter um sistema pronto em seis meses, o que deixa os envolvidos/pessoas satisfeitos pela sensação de atingimento de meta em um prazo razoável.

Embora os sistemas atendidos neste modelo de desenvolvimento tenham características muito particulares, devido às características da tecnologia, o desenvolvimento com a tecnologia APEX é uma solução que consegue atender a área fim do órgão, que vinha sendo pouco servida pela TI. Há muitos anos essas unidades vinham usando apenas o sistema processual da casa. A partir do momento que eles tiveram acesso a uma ferramenta que lhes

deu certa independência, rapidamente aderiram. Antes disso, a TI nem conseguia atender esse tipo de demanda.

#### **4.2.5 Lições Aprendidas**

Para uma área de negócio, que demandava novos sistemas e correções muito rápidas, a TI optou por manter um servidor da TI alocado no departamento dedicado às aplicações EUD desta unidade. Segundo o Órgão Público B, esta abordagem é melhor do que ter um estagiário lidando diretamente com chefes de setor. Além disso, estagiários nem sempre conseguem resolver problemas quando há urgência.

Com relação à tecnologia, para o órgão, o APEX pode não parecer tão amigável para o usuário não ligado a tecnologia. Além disso, à medida que as interfaces se tornam mais complexas, o desenvolvimento se torna mais lento. Outra limitação do APEX é em relação aos relatórios que nem sempre ficam bons no Apex, nestes casos o Órgão Público B opta por outras ferramentas especializadas.

#### **4.2.6 Perspectivas Futuras**

O Órgão Público B espera para o futuro criar um fluxo contínuo de estagiários, aos moldes do Órgão Público A, para apoiar o desenvolvimento EUD.

Eles consideram importante também realizar uma espécie de campanha de incentivo para sensibilizar e motivar os servidores para o desenvolvimento EUD nas pontas, garantindo um certo nível de qualidade. Apontam que há uma questão de motivação intrínseca dos servidores para se engajarem neste modelo, além da necessidade de reconhecimento pelo trabalho e benefício trazidos para a área.

Outro ponto citado para o futuro é o investimento em banco de dados e infraestrutura.

### **4.3 Estudo de Caso C**

O Órgão Público C é um órgão integrante do Poder Judiciário que conta com 10 mil servidores. Na área de TI trabalham 151 servidores, sendo 65 com análise de sistemas e 86 com programação; além de 3 estagiários.

Em meados de 2013 e 2014, um Subsecretário de Relacionamento e Atendimento da TI do Órgão Público C conheceu a estratégia de desenvolvimento descentralizado baseada na abordagem EUD do Órgão Público A, cujos sistemas são desenvolvidos por servidores e estagiários, utilizando a Ferramenta APEX, da Oracle.

Neste mesmo momento, o Órgão Público C estava adquirindo o pacote Oracle com a possibilidade de utilização do APEX e algumas unidades do Órgão Público C se interessaram pelo desenvolvimento descentralizado. Era de conhecimento que ocorria no órgão o

desenvolvimento pontual e informal de aplicações nas áreas de negócio, caracterizado como uma “TI cinza”, utilizando diferentes ferramentas como *Microsoft Access, Excel e SharePoint*.

Após alguns anos de utilização do APEX, o setor de infraestrutura começou a reportar problemas de demanda pelo fato de que não conseguiam dar o suporte necessário, o que motivou o Órgão Público C a visitar novamente o Órgão Público A, a fim de coletar informações sobre as estratégias de EUD/descentralização empregadas.

Em seguida, o Órgão Público C buscou formalizar o desenvolvimento descentralizado. Uma Portaria foi publicada regulamentando o *Desenvolvimento Descentralizado* como estratégia de desenvolvimento departamental. A Portaria define questões como: padrões do desenvolvimento, questões operacionais, envolvimento de estagiários e responsáveis na área de negócio, e a responsabilização das áreas de negócio.

#### **4.3.1 Estratégia de adoção do EUD: áreas do negócio, critérios para seleção das soluções, suporte dado pela TI**

Dada a política de desenvolvimento, toda demanda passa pelo fluxo de desenvolvimento (centralizado ou descentralizado). O Comitê de TI recebe as demandas de desenvolvimento de aplicações e as classifica como corporativo, que não são adequados ao EUD, ou como departamental, que podem ser desenvolvidas no modelo descentralizado/EUD. A distinção entre corporativo e departamental acontece de forma clara no Órgão Público C.

Cada demanda por desenvolvimento de aplicações pelas unidades segue um fluxo definido. Quando uma unidade se interessa pelo desenvolvimento descentralizado/EUD, ela submete a intenção à Assessoria de Governança de TI e de Segurança de Informação, que faz uma análise prévia com as unidades de TI e submete a intenção ao Comitê de Governança de TI, que autoriza o desenvolvimento ou não da aplicação EUD mediante os requisitos que devem ser atendidos.

A governança de TI possui três estagiários da área de TI. Embora seja possível contratar estagiários para o desenvolvimento descentralizado por um plano jurídico-administrativo, a TI não tem avaliado a viabilidade da contratação de estagiários para este fim. A Portaria estabelece que a unidade interessada deve nomear dois servidores para garantir o desenvolvimento da aplicação. Esta medida visa minimizar a dependência dos estagiários e reduzir o impacto da rotatividade dos estagiários. Os servidores nomeados são responsáveis pela aplicação, independente dos estagiários. Vale ressaltar que não há resistência por parte dos servidores em adotar o EUD. Esses reconhecem que há benefícios, e percebem que a utilização de padrões de desenvolvimento é melhor que o desenvolvimento aleatório e descontrolado.

### 4.3.2 Tecnologias Adotadas

A ferramenta principal escolhida pelo Órgão Público C para ser trabalhada no modelo EUD é o Apex, da Oracle. O treinamento para a utilização da ferramenta foi ministrado utilizando o curso online de capacitação do Órgão Público A. Os servidores a serem nomeados são avaliados quanto a conhecimentos mínimos em MER e SQL. Além do Apex, também utilizam *SharePoint* e *QlikView* (ferramenta de BI).

### 4.3.3 Qualidade e Segurança

Formalmente, não há uma política ou avaliação de qualidade das aplicações produzidas neste modelo. A partir da Portaria publicada, cada aplicação desenvolvida no modelo Desenvolvimento Descentralizado é departamental e de responsabilidade da unidade que a produziu. A área de TI age como uma corresponsável. A área de TI fornece suporte de infraestrutura e pretende no futuro padronizar o desenvolvimento (nomenclatura de tabelas, por exemplo). A Portaria impede que novas ferramentas de desenvolvimento sejam utilizadas sem estarem devidamente homologadas.

Cada unidade é responsável por definir as regras de negócio e as especificações da aplicação. Aspectos como: negócio, segurança, backup e acesso a dados sigilosos são do escopo da unidade. Em casos críticos (os servidores deixaram a unidade), a unidade de governança dá suporte à aplicação da unidade.

Por questões de segurança das informações do órgão e para manter a integridade do servidor de banco de dados há um controle de acesso aos dados (leitura, escrita e etc.), evitando consultas que possam sobrecarregar ou derrubar o serviço. As aplicações departamentais possuem permissão de escrita e leitura às suas próprias tabelas e somente leitura caso seja autorizado acessos externos no seu ambiente de dados.

### 4.3.4 Impactos e Benefícios

Atualmente, há um ambiente mais previsível para as unidades. Há um início de "desafogamento" da fila de demandas por desenvolvimento de aplicações. As demandas represadas diminuíram e a percepção das unidades em relação à TI está melhorando.

Percebe-se uma satisfação dos servidores que conseguem responder melhor, dada a diminuição das demandas. E percebe-se a satisfação dos servidores das unidades por estarem construindo o que eles necessitam sem precisar esperar a área de TI realizar, os quais encaram a atividade como um "desafio pessoal".

Há diversos exemplos de impactos positivos no órgão. Há sistemas não corporativos desenvolvidos por servidores que se mostraram de grande valia, aumentando a percepção da capacidade do modelo descentralizado/EUD.

Um impacto, no qual não se julga como negativo pela TI, da nomeação/adoção dos servidores pelas áreas de negócio é a evasão de servidores da área de TI (centralizada). Unidades de negócio têm convidado servidores da área de TI, ofertando cargos comissionados, para desenvolverem em EUD na unidade.

Apesar do ganho de autonomia, as áreas de negócios têm consumido mais recursos de TI. Uma preocupação é que a infraestrutura inicial oferecida seja insuficiente para conter o crescimento constante da demanda das unidades interessadas no desenvolvimento descentralizado. Isso poderá ocasionar o não atendimento às expectativas do EUD e gerar uma “Fila dos Descentralizados”.

#### **4.3.5 Lições Aprendidas**

A oportunidade de poder se inspirar e até se apoiar em um órgão que adotou previamente a estratégia foi de grande valia.

Dada a falta de experiências em produção descentralizada, o órgão espera que lições valiosas possam ser adquiridas ao decorrer do processo de implantação da abordagem de desenvolvimento descentralizado.

#### **4.3.6 Perspectivas Futuras**

O Órgão Público C prevê a padronização e a estruturação da área de desenvolvimento descentralizado/EUD. Além de evoluir os parâmetros de seleção do desenvolvimento centralizado e descentralizado.

### **4.4 Estudo de Caso D**

O Órgão Público D é do Poder Administrativo Federal, trata de assuntos financeiros e contábeis da federação. Conta com um quadro de aproximadamente 600 servidores, sendo que destes, aproximadamente 60 estão na área de TI.

#### **4.4.1 Histórico do EUD**

O EUD no Órgão Público D foi arquitetado em meados de 2011, quando dois membros do Órgão, durante uma visita ao Órgão Público A, conheceram o desenvolvimento EUD. Com a intenção de implantá-lo, o Órgão Público D buscou uma ferramenta que possibilitasse o desenvolvimento mais rápido que o desenvolvimento tradicional, encontrando a ferramenta Oracle/Apex. Posteriormente, um pequeno grupo de funcionários da TI foi alocado para conhecer a ferramenta e desenvolver os primeiros sistemas nesta tecnologia.

Ao longo de quatro anos, o grupo de funcionários da TI trabalhou com desenvolvimento de soluções de software para a área de TI e para as áreas de negócio. Durante este período, a complexidade dos sistemas desenvolvidos aumentava de forma gradativa. Um processo de

desenvolvimento com base na metodologia ágil foi criado e refinado e atualmente o Órgão considera o seu processo robusto e completo, sendo utilizado pelos integrantes do desenvolvimento de soluções de TI.

Durante o ano de 2016, foi planejada a primeira experiência do EUD com uma área de negócio. O projeto piloto começou em Novembro do mesmo ano. No entanto, em Janeiro de 2017, tendo sido detectado que a tecnologia não era adequada para as necessidades da aplicação demandada, o projeto foi suspenso. O Órgão D atualmente busca novas demandas das áreas de negócio que possam ser tratadas pela ferramenta de desenvolvimento EUD disponível no órgão.

#### **4.4.2 Estratégia de adoção do EUD: áreas do negócio, critérios para seleção das soluções, suporte dado pela TI**

Para que um sistema seja desenvolvido no modelo EUD, há alguns critérios a serem obedecidos, dentre eles:

- O sistema a ser desenvolvido deve ser departamental, ou seja, será utilizado apenas por uma área de negócio;
- O sistema deverá ser de baixa complexidade, para isso não poderá realizar integração com outras aplicações, nem poderá ter dependências;
- O usuário desenvolvedor deve pertencer à área de negócio para a qual a aplicação será desenvolvida.

No momento em que uma área de negócio demandar o desenvolvimento de uma aplicação no modelo EUD seguindo os critérios mencionados, uma pessoa da área de negócio deverá ser indicada para se tornar um desenvolvedor EUD. Essa pessoa então passará, primeiramente, por um treinamento realizado pela equipe de TI, durante o qual ela irá aprender sobre a ferramenta e normas de desenvolvimento.

A área de TI fornecerá suporte ao usuário desenvolvedor por meio de um servidor capacitado para sanar dúvidas técnicas, durante o desenvolvimento da aplicação.

#### **4.4.3 Tecnologias Adotadas**

A ferramenta escolhida para a abordagem EUD foi o *Apex*, da *Oracle*. Essa escolha foi devida ao fato de que o Órgão Público D já possui a licença *Oracle*, não sendo necessários novos investimentos. Não foi realizado um levantamento de outras ferramentas candidatas.

Atualmente a TI do Órgão D desenvolve aproximadamente 90% das aplicações em *Apex* e os outros 10% em outras tecnologias como Java, Check Point, entre outros.

#### **4.4.4 Qualidade e Segurança**

A qualidade e a segurança das aplicações ainda não podem ser medidas, pois até este momento, nenhuma aplicação foi desenvolvida pela abordagem EUD. No entanto, o Órgão D acredita que a responsabilidade da aplicação, bem como a garantia de aspectos como qualidade e segurança devem ser claramente evidenciadas para o desenvolvedor antes do início do processo de desenvolvimento.

#### **4.4.5 Impactos e Benefícios**

Dado que a estratégia de desenvolvimento descentralizado, adotando a tecnologia Apex, ainda não foi testada dentro da área de negócio, não houve nenhum impacto ou benefício identificado pela TI uma vez que não foi implantada. Entretanto, o Órgão espera que a área de TI consiga diminuir a demanda de sistemas de baixa complexidade pelas áreas de negócio quando a estratégia for implantada.

#### **4.4.6 Lições Aprendidas**

O Órgão Público D espera obter bons resultados ao inserir o desenvolvimento descentralizado nas áreas de negócio.

A área de TI notou que a motivação do usuário final desenvolvedor deve ser cuidadosamente avaliada, uma vez que ele pode entender que a área de TI está terceirizando seu serviço para a área de negócio com o desenvolvimento descentralizado.

Outro aspecto que o Órgão D ressalta é a importância do princípio da prudência comum à área contábil. Neste sentido, a TI acredita que a implantação do EUD deve ser cautelosa e bem avaliada.

#### **4.4.7 Perspectivas Futuras**

O órgão tem como expectativa executar um projeto piloto para refinar o modelo de desenvolvimento descentralizado e avaliar os resultados.

## **5 ANÁLISE DE ESTRATÉGIAS DE DESENVOLVIMENTO DESCENTRALIZADO EM EMPRESAS PRIVADAS**

Nesta seção apresentamos alguns casos de desenvolvimento de software através de ferramentas apropriadas para desenvolvimento por usuários finais. É importante ressaltar que as estratégias de cada organização variam de acordo com a motivação do uso de tais ferramentas. Até o momento, não tivemos acesso aos detalhes dos mecanismos de descentralização e do perfil dos usuários desenvolvedores.

### **5.1 Caso Union Investment e APEX**

A Union Investment é uma empresa alemã de consultoria financeira, fundada em 1956, atualmente é uma das maiores empresas da Europa de consultoria financeira com mais de 285 bilhões de euros de recursos gerenciados, sendo aproximadamente 165,5 bilhões de euros de recursos de clientes institucionais.

#### **5.1.1 Histórico**

"Temos uma longa história com o Oracle DataBase, que confiamos para ser eficiente em nosso negócio de gerenciamento de ativos. É certamente a base de dados líder do mercado e, usando o Oracle Application Express, desenvolvemos uma aplicação web que muda o jogo no cerne do nosso negócio, o que permitiu melhorias nos processos ao longo de nossas operações principais" Rolf Fillinger, Líder do Time, (UNION INVESTMENT, 2016).

#### **5.1.2 Estratégias**

De acordo com a Oracle (UNION INVESTMENT, 2016), sobre a Union Investment, em vez de expandir seu sistema existente, ela decidiu introduzir um novo sistema com as interfaces necessárias e, em seguida, executar uma migração de dados.

Na primeira fase do projeto, todos os novos diálogos foram fornecidos de forma simplificada, onde foram coordenados com os departamentos das partes interessadas. Seguindo especificações finais, a segunda fase resolveu questões como autorizações, visões internas e externas e interfaces, e envolveu mais de 270 exames de peritos, vendo o projeto através de uma produção bem-sucedida (UNION INVESTMENT, 2016).

#### **5.1.3 Tecnologias Adotadas**

A ferramenta escolhida pela Union Investment foi o Oracle Application Express. Essa escolha é explicada por Rolf Fillinger, Líder do Time,

“Já tínhamos uma boa experiência com o Oracle Application Express no passado, especialmente desenvolvendo aplicativos web individualizados e facilmente ajustáveis, armazenados no Oracle Database, e também estabelecemos um bom relacionamento com o Oracle Partner MT AG. No entanto, pesquisamos no mercado para ver se havia um produto disponível que poderia atender o vasto escopo de nossos requisitos de projeto de gerenciamento de fundos, mas não havia nenhum. Quando decidimos desenvolver nossa própria aplicação, precisávamos escolher entre baseada em Java e baseada em navegador. Escolher Java significaria uma versão de custo total, full-client para 800 pessoas e, portanto, uma solução baseada em navegador era preferível, e nesta fase apenas o Oracle APEX poderia atender às nossas necessidades.” (UNION INVESTMENT, 2016)

#### **5.1.4 Necessidades**

As necessidades do contexto descritos pela Oracle (UNION INVESTMENT, 2016) sobre a Union Investment no ano de 2015 consistiam de:

- Aumentar a eficiência operacional ao permitir acesso aos dados de fundos da empresa para mais de 800 funcionários em um único sistema;
- Permitir um gerenciamento estruturado de dados de aproximadamente 1000 fundos de investimento, de forma que os gerentes de portfólio não necessitem se envolver com a TI para configuração e modificação de fundos;
- Melhorar a competitividade ao aumentar a velocidade e a eficiência na criação e gerenciamento de fundos de investimento;
- Desenvolver uma aplicação intuitiva de gerenciamento de fundos de forma que os gerentes de fundos possam facilmente projetar e lançar novos fundos de investimentos.

#### **5.1.5 Solução**

A solução, chamada de FondsProfiler, foi desenvolvida utilizando Oracle Application Express 4.1 (Oracle Apex) e é conhecida como o maior sistema já desenvolvido na ferramenta. O desenvolvimento da FondsProfiler teve uma duração de 12 meses e consiste em 67 páginas, 344 tabelas, 366 views e 107 pacotes.

Contou com o apoio de MT AG, empresa parceira da Oracle. Tal sistema, segundo o estudo de caso, cumpre satisfatoriamente as necessidades da Union Investment e atualmente continua utilizando da ferramenta para a construção de sistemas internos da empresa.

## 5.2 Caso Liberty Seguros e OutSystems

Liberty Seguros é uma empresa americana de seguros, parte do conglomerado Liberty Mutual Group, uma seguradora de nível global que oferece um grande alcance de produtos de seguro e serviços e emprega mais de 38 mil funcionários em aproximadamente 900 escritórios em todo o mundo.

### 5.2.1 Histórico

Começou como uma única prova de conceito para uma aplicação clínica e ao longo da última década tem crescido para se tornar uma das mais longas e mais complexas instalações já realizadas por um cliente OutSystems. A Liberty Seguros é um bom exemplo dos benefícios a longo prazo oferecidos pela OutSystems Platform (LIBERTY SEGUROS, 2016).

### 5.2.2 Estratégias

Faz 10 anos que a primeira aplicação em Liberty Connect foi lançada. Essa solução de processo clínico ainda está viva hoje, excedendo a expectativa inicial de quatro anos de vida. Esta solução foi desenvolvida e implantada em três meses por uma equipe de três desenvolvedores e hoje, a TI ainda é capaz de oferecer novos recursos (LIBERTY SEGUROS, 2016).

O Liberty Connect é uma plataforma colaborativa madura com os clientes no seu núcleo, permitindo que todas as áreas da empresa, incluindo parceiros externos, interajam em tempo real. Dessa forma, foi possível para a Liberty sustentar completamente seus negócios sem perder a capacidade de oferecer serviços centrados no cliente, tornando a aplicação, extremamente importante para a estratégia da empresa (LIBERTY SEGUROS, 2016).

Dessa forma, fazem parte das estratégias da Liberty, os seguintes tópicos (LIBERTY SEGUROS, 2016):

- Todas as unidades de negócio e parceiros externos conectados através de uma única plataforma.
- A plataforma OutSystems é usada para todo o desenvolvimento personalizado.
- O primeiro aplicativo implantado ainda é executado hoje (Aumento do tempo de vida das aplicações).
- Sistemas como Saperian, Sharepoint e Dynamics substituídos por Liberty Connect.

### 5.2.3 Tecnologias Adotadas

Liberty Seguros utiliza da ferramenta OutSystems por mais de 10 anos lançando ao todo 39 aplicações. Essa escolha foi feita pois (LIBERTY SEGUROS, 2016),

- Lhes permitiria integrar e ampliar seus sistemas de seguro back-end existentes e criar front-ends especificamente adaptados ao seu modelo de negócio inovador
- A flexibilidade para manter e alterar aplicativos durante a execução garantiria que as aplicações poderiam ser atualizadas, aumentando assim a sua vida útil média;
- Forneceu a oportunidade de criar parcerias com usuários corporativos para criar rapidamente aplicações departamentais para atender necessidades imprevistas ou urgentes.

### 5.2.4 Necessidades

As necessidades que levaram ao uso da ferramenta envolviam conectar cada departamento e parceiro externo através de uma solução única que permitia tal colaboração, incluindo os próprios clientes da Liberty Seguros (LIBERTY SEGUROS, 2016).

Outras necessidades eram aumentar a vida média das aplicações para mais de quatro anos e fazer a TI apoiar as necessidades táticas de negócios através da criação de micro aplicações departamentais (LIBERTY SEGUROS, 2016).

### 5.2.5 Solução

O sistema desenvolvido pela empresa, chamado de Liberty Connect, atingiu satisfatoriamente os resultados esperados. Atualmente, a equipe de TI da Liberty Seguros mantém um dos mais complexos ecossistemas OutSystems do mundo (LIBERTY SEGUROS, 2016).

Dessa forma, esse sistema é uma solução personalizada que gerencia múltiplos processos de negócios. Ele, hoje, suporta mais de 4000 usuários internos e externos, roda em um cluster de nove front-ends e inclui integração com vários sistemas de registro (LIBERTY SEGUROS, 2016).

## 6 REFERÊNCIAS

ALAVI, M.; WEISS, I. R. Managing the Risks Associated with End-User Computing, Journal of Management Information Systems. v. 2, p. 5–20, 2015.

AMOROSO, Donald L. Organizational Issues of End-User Computing. ACM SIGMIS Database, v. 19, n. 3-4, p. 49-58, 1988.

AMOROSO, Donald L. Using End User Characteristics to Facilitate Effective Management of End User Computing. Journal of Organizational and End User Computing (JOEUC), v. 4, n. 4, p. 5-16, 1992.

AMOROSO, Donald L., CHENEY, Paul H. Quality end user-developed applications: Some essential ingredients. ACM SIGMIS Database, 23(1), 1-11, 1992.

BANERJEE, Sean, et al. Log-Based Reliability Analysis of Software as a Service (SaaS). In: IEEE, 21. 2010, San Jose, CA. Software Reliability Engineering (ISSRE). San Jose, CA: [s.n.], 2010. p. 239-248. Disponível em: <<http://ieeexplore.ieee.org/document/5635046/>>. Acesso em: 21 nov. 2016.

BAKER, Kevin et al. Empirical development of a heuristic evaluation methodology for shared workspace groupware. ACM conference on Computer supported cooperative work, New Orleans, Louisiana, USA, p. 96-105, jun. 2011. Disponível em: <<http://doi.acm.org/10.1145/587078.587093>>. Acesso em: 28 nov. 2016.

BARKER, Sandra. End User Computing and End User Development: Exploring Definitions for the 21st Century. Tese de Doutorado. IGI Publishing. 2007.

BARKER, Sandra Kay; FIEDLER, Brenton. Developers, Decision Makers, Strategists or Just End-users? Redefining End-User Computing for the 21st Century: A Case Study. Tese de Doutorado. IGI Global. 2011.

BARKI, H., & HARTWICK, J. Rethinking the concept of user involvement. MIS Quarterly, 13 (March), 52-63, 1989.

BRANCHEAU, James C.; BROWN, Carol V. The management of end-user computing: status and directions. ACM Computing Surveys (CSUR), v. 25, n. 4, p. 437-482, 1993.

BRASIL, Ministério do Planejamento, Desenvolvimento e Gestão. Edital do Pregão Eletrônico para Registro de Preços nº 18/2016. 2016. Disponível em: <[http://www.planejamento.gov.br/aceso-a-informacao/licitacoes-e-contratos/licitacoes/pregao/2016/16\\_lici\\_pregao18\\_edital.pdf](http://www.planejamento.gov.br/aceso-a-informacao/licitacoes-e-contratos/licitacoes/pregao/2016/16_lici_pregao18_edital.pdf)> Acesso em: 13 de dez. de 2016.

BOEHM, B. W. et al. Characteristics of Software Quality. Amsterdam: North-Holland, 1978.

BURNETT, Margaret et al. Scaling up visual programming languages. *IEEE Computer*, 28(3), 45, 1995.

CALE, E. G. Quality issues for end-user developed software. *Journal of Systems Management*. p. 36–39, 1994.

CAVANO, J. P., & MCCALL, J. A. A framework for the measurement of software quality. *Proceedings of the Software Quality and Assurance Workshop*, 133-140, 1978.

CHAFFEY, D., & Wood, S. (2005). *Business Information Management Improving Performance Using Information System*. Pearson Education Limited, Essex, England.

CHAN, Yolande E.; STOREY, Veda C. The use of spreadsheets in organizations: Determinants and consequences. *Information & Management*, v. 31, n. 3, p. 119-134, 1996.

CHENEY, P. H., MANN, R. I., AMOROSO, D. L. Organizational factors affecting the success of end-user computing. *Journal of Management Information Systems*, 3(1), 65-80, 1986.

COMELLA-DORDA, S. , J.C. Dean, E. Morris, P. Oberndorf. A Process for COTS Software Product Evaluation. Springer-Verlag, ICCBSS 2002, LNCS 2255, pp.86–96.

COSTABILE, M.F., Fogli, D., Mussio, P. & Piccinno, A. (2005). A meta-design approach to End-user development, in *Proceedings of the 2005 IEEE Symposium on Visual Languages and Human-Centric Computing*. IEEE Computer Society, 308-310.

COX P. T.; GILES F. R.; PIETRZYKOWSKI T. Prograph: a step towards liberating programming from textual conditioning. *IEEE Workshop on Visual Languages*, pp. 150-156, 1989.

CRAMM, Susan. IT Centralization or Decentralization? *Harvard Business Review*. July 2008. <https://hbr.org/2008/07/it-centralization-or-decentral>

CRUZ, Cláudio Silva da; ANDRADE, Edméia Leonor Pereira de; FIGUEIREDO, Rejane Maria da Costa. *Processo de Contratação de Serviços de Tecnologia da Informação para Organizações Públicas*. Brasília, DF, 2010. Disponível em: <[http://www.mct.gov.br/upd\\_blob/0216/216672.pdf](http://www.mct.gov.br/upd_blob/0216/216672.pdf)>. Acesso em: 12 dez. 2016.

DANN, W.; COOPER, S.; PAUSCH, R. *Learning to program with Alice*. Prentice Hall, 2006.

DAVIS, G. B.; M. H. OLSON. Management Information Systems: Conceptual Foundations, Structure, and Development. 2nd Edition, McGraw-Hill, New York, 1985.

DAVIS, J.; SRINIVASAN, A. Incorporating user diversity into information system assessment. In N. BjornAndersen & G. Davis (Eds.), Information Systems Assessment, New York: Knowledge Industry, pp. 83-98, 1988.

DOLL, W. J.; TORKZADEH, G. The measurement of end-user computing satisfaction. MIS Quarterly, 12(2), 259-274, 1988.

DOLL, W. J.; TORKZADEH, G. A discrepancy model of end-user computing involvement. Management Science. v. 35(10), p. 1151–1171, 1989.

DOWNEY, James P.; BARTCZAK, Summer. End User Computing Research Issues and Trends. Advanced Topics in End User Computing, Volume 4, v. 4, p. 1, 2005.

FISCHER, Gerhard. Domain-oriented design environments. Automated Software Engineering 1(2), pp. 177-203, 1994.

FISCHER, G., GIACCARDI, E., Ye, Y., SUTCLIFFE, A.G., and MEHANDJIEV, N. Meta-design: a manifesto for end-user development. Commun. ACM 47, 9 (2004), 33–37.

FOWLER, Martin; SADALAGE, Pramod. Evolutionary Database Design, 2016. Disponível em: <<https://martinfowler.com/articles/evodb.html>>. Acesso em: 12 dez. 2016.

GARAVAN, Thomas N.; MCCRACKEN, Clare. Introducing end-user computing: The implications for training and development-Part 1. Industrial and Commercial Training, v. 25, n. 9, 1993.

GHAZARIAN, Armen. How to Conduct a Usability Heuristic Evaluation. 2014. DesignModo. Disponível em: <<https://designmodo.com/usability-heuristic-evaluation/>>. Acesso em: 24 out. 2016.

GREEN, T. R. G.; Petre, M. Usability analysis of visual programming environments: A cognitive dimensions framework. Journal of Visual Languages and Computing 7(2), 131–174, 1996.

HAUSAWI, Yasser M . 2016. Current Trend of End-Users’ Behaviors Towards Security Mechanisms. In: Human Aspects of Information Security, Privacy, and Trust, pp.140-151.

HUITFELDT, B., & MIDDLETON, M. The assessment of software quality from the user perspective: Evaluation of a GIS implementation. *Journal of End User Computing*, 13(1), 3-11, 2001.

IACOB, Claudia. Design Patterns in the Design of Systems for Creative Collaborative Processes. Proceedings of the Third International Conference on End-user Development, Torre Canne, Italy, p. 359-362, jun. 2011. Disponível em: <<http://dl.acm.org/citation.cfm?id=2022939.2022995>>. Acesso em: 28 nov. 2016.

INTERNATIONAL ORGANIZATION FOR STANDARDIZATION. ISO/IEC 25000: Systems and software engineering – Systems and software Quality Requirements and Evaluation (SQuaRE) – Measurement of system and software product quality. Geneva, 2011.

INTERSYSTEMS. Evaluating Integration Software. Cambridge: Intersystems Ensemble, 2007. Disponível em: <<http://www.intersystems.com/assets/evaluating-integration-software.pdf>>. Acesso em: 24 out. 2016.

IVES, B., S. HAMILTON, and G. B. DAVIS. “A Framework for Research in Computer-Based Management Information Systems”. *Management Science*, Sept. 1980, pp. 910 - 934.

JADHAV, Anil S.; SONAR, Rajendra M.. 2009. Evaluating and selecting software packages: A review. *Inf. Softw. Technol.* 51, 3 (March 2009), 555-563.

JAWAHAR, I.M. & ELANGO, B. (2001). The effects of attitude, goal setting and self-efficacy on end user performance. *Journal of End User Computing*, 13(2), 40-45.

KELLEHER, C.; PAUSCH, R. Lessons learned from designing a programming system to support middle school girls creating animated stories. *IEEE Symposium on Visual Languages and Human-Centric Computing*, pp. 165-172, 2006.

KELLEHER, C.; PAUSCH, R.; KIESLER, S. Storytelling Alice motivates middle school girls to learn computer programming. *ACM Conference on Human Factors in Computing Systems*, San Jose, California, pp. 1455-1464, 2007.

KLANN, Markus; PATERNÒ, Fabio; WULF, Volker. Future perspectives in end-user development. In: *End user development*. Springer Netherlands, p. 475-486. 2006.

KO, Andrew J. et al. The state of the art in end-user software engineering. *ACM Computing Surveys (CSUR)*, v. 43, n. 3, p. 21, 2011.

LEWIS, G. A. (2010). Emerging Technologies for Software - Reliant Systems of Systems. SEI – Software Engineering Institute. Technical Note CMU/SEI-2010-TN-019. Pp 15. Disponível em [http://resources.sei.cmu.edu/asset\\_files/TechnicalNote/2010\\_004\\_001\\_15182.pdf](http://resources.sei.cmu.edu/asset_files/TechnicalNote/2010_004_001_15182.pdf)

LIBERTY SEGUROS. Liberty Insurance Streamlines and Automates End-to-end Business Processes for Thousands of Customers. Disponível em: <<https://www.outsystems.com/case-studies/liberty/insurance-management-system/>>. Acesso em: 13 dez. 2016.

LIEBERMAN, Henry. Your Wish is my Command: Programming by Example. San Francisco: Morgan Kaufmann, 2001.

LIEBERMAN, Henry et al. End-user development: An emerging paradigm. In: End user development. Springer Netherlands, p. 1-8. 2006.

LIN, W. T.; SHAO, B. B. M. The relationship between user participation and system success: A simultaneous contingency approach. Information & Management. v. 37, p. 283–295, 2000.

MARTIN, E., Brown, C., DeHayes, D., Hoffer, J., & Perkins, W. (2005). Managing Information Technology. Pearson Education.

MCGILL, Tanya. The Effect of End User Development on End User Success. Advanced Topics in End User Computing, Volume 4, v. 4, p. 21, 2004.

MCGILL, T. J. et al. User satisfaction as a measure of success in end user application development: An empirical investigation. In M. Khosrowpour (Ed.), Proceedings of the 1998 IRMA Conference, Hershey, USA: Idea Group Publishing, pp. 352-357, 1998.

MEHANDJIEV, Nikolay; SUTCLIFFE, Alistair; LEE, Darren. Organizational view of end-user development. In: End user development. Springer Netherlands, p. 371-399. 2006.

MIKA, Shelley. Five Steps to Database Integration, 2006. Disponível em: <<http://www.government-fleet.com/article/story/2006/03/five-steps-to-database-integration.aspx>>. Acesso em: 12 dez. 2016.

MINTZBERG, Henry. Structure in 5's: A Synthesis of the Research on Organization Design. Management science, v. 26, n. 3, p. 322-341, 1980.

MINTZBERG, Henry. Organization Design: Fashion or Fit? Harvard Business Review. January 1981 issue. <https://hbr.org/1981/01/organization-design-fashion-or-fit>.

MITRE CORPORATION. Systems Integration: Identify and Assess Integration and Interoperability (I&I) Challenges. In: MITRE CORPORATION (Usa). Systems Engineering Guide. Bedford: Mitre, 2014. p. 381-385. Disponível em: <<https://www.mitre.org/publications/systems-engineering-guide/se-lifecycle-building-blocks/systems-integration/identify-and-assess-integration-and-interoperability-iampi-challenges>>. Acesso em: 24 out. 2016.

MITRE CORPORATION. Systems Integration: Develop and Evaluate Integration and Interoperability (I&I) Solution Strategies. In: MITRE CORPORATION (Usa). Systems Engineering Guide. Bedford: Mitre, 2014. p. 386-389. Disponível em: <<https://www.mitre.org/publications/systems-engineering-guide/se-lifecycle-building-blocks/systems-integration/identify-and-assess-integration-and-interoperability-iampi-challenges>>. Acesso em: 24 out. 2016.

MØRCH, Anders I. Method and tools for tailoring of object-oriented applications: an evolving artifacts approach. University of Oslo, Department of Informatics, 1997.

NARDI, Bonnie A.; MILLER, James R. An ethnographic study of distributed problem solving in spreadsheet development. In: Proceedings of the 1990 ACM conference on Computer-supported cooperative work. ACM, p. 197-208. 1990.

NIELSEN, Jakob. 10 Usability Heuristics for User Interface Design. 1995. Disponível em: <<https://www.nngroup.com/articles/ten-usability-heuristics/>>. Acesso em: 24 out. 2016.

NIELSEN, Jakob. How to Conduct a Heuristic Evaluation. 1995. Disponível em: <<https://www.nngroup.com/articles/how-to-conduct-a-heuristic-evaluation/#>>. Acesso em: 24 out. 2016.

NIELSEN, Jakob. Usability 101: Introduction to Usability. 2012. Disponível em: <<https://www.nngroup.com/articles/usability-101-introduction-to-usability/>>. Acesso em: 14 dez. 2016.

NIESS, M.; SADRI, P.; AND LEE, K. Dynamic Spreadsheets as Learning Technology Tools: Developing Teachers' Technology Pedagogical Content Knowledge (TPCK). American Educational Research Association, 2007.

O'DONNELL, D., MARCH, S. End user computing environments—Finding a balance between productivity and control. Information & Management, 13(1), pp. 77-84, 1987.

ORACLE. Oracle Store: Oracle Database Enterprise Edition. Disponível em: <[https://shop.oracle.com/apex/f?p=DSTORE:6:::NO:RP,6:P6\\_LPI,P6\\_PROD\\_HIER\\_ID:4509382199341805719938,4509958287721805720011](https://shop.oracle.com/apex/f?p=DSTORE:6:::NO:RP,6:P6_LPI,P6_PROD_HIER_ID:4509382199341805719938,4509958287721805720011)>. Acesso em: 24 out. 2016.

ORACLE Application Express, ORACLE, 2016. Disponível em: <<https://apex.oracle.com/>>. Acesso em: 13 dez. 2016.

OUTSYSTEMS, OUTSYSTEMS, 2016. Disponível em: <<https://www.outsystems.com/>>. Acesso em: 13 dez. 2016.

PANKO, R. R.; HALVERSON, R. P. Spreadsheets on trial: A survey of research on spreadsheet risks. Proceedings of the Twenty-Ninth Hawaii International Conference on System Sciences, 2, 326-335, 1996.

PARRISH, Meagan. 6 Ways to Evaluate Integration Capabilities. IntraLinks Blog. Disponível em: <<https://blogs.intralinks.com/2014/05/6-ways-evaluate-integration-capabilities/>>. Acesso em: 24 out. 2016.

PATERNÒ, Fabio. End User Development: Survey of an Emerging Field for Empowering People. ISRN Software Engineering, v. 2013, 2013.

PIEROTTI, Deniese. Heuristic Evaluation - A System Checklist. 2006. Disponível em: <<ftp://cs.uregina.ca/pub/class/305/lab2/example-he.html>>. Acesso em: 24 out. 2016.

RAINER, R. Kelly; HARRISON, Allison W. Toward development of the end user computing construct in a university setting. Decision Sciences, v. 24, n. 6, p. 1187-1202, 1993.

REPENNING, Alexander. Agentsheets: A Tool for Building Domain Oriented-Dynamic Visual Environments. Technical Report, Dept of Computer Science, CU/CS/693/93. Boulder, CO: University of Colorado, 1993.

RICHARDSON, Clay; Rymer, John R. The Forrester Wave™: Low-Code Development Platforms. Q2 2016 The 14 Providers That Matter Most And How They Stack Up, FORRESTER, Abril, 2016.

RICHARDSON, Clay; Rymer, John R. Vendor Landscape: The Fractured, Fertile Terrain of Low-Code Application Platforms. FORRESTER, Janeiro, 2016.

RITTENBERG, Larry E.; SENN, Ann. End-user computing. Internal Auditor, v. 50, n. 1, p. 35-40, 1993.

ROCKART, J.F.; FLANNERY, L.S. The Management of End User Computing. Communications of the ACM , Vol. 26, No. 10, 1983, p. 776-784.

SALESFORCE CRM, SALESFORCE, 2016. Disponível em: <<https://www.salesforce.com>>. Acesso em: 13 dez. 2016.

SALESFORCE Lightning, SALESFORCE, 2016. Disponível em: <<https://developer.salesforce.com/lightning>>. Acesso em: 13 dez. 2016.

SALESFORCE Mobile, SALESFORCE, 2016. Disponível em: <<https://www.salesforce.com/products/platform/solutions/mobile/>>. Acesso em: 13 dez. 2016.

SCAFFIDI, C.; SHAW, M.; MYERS, B. Estimating the Numbers of End Users and End User Programmers. IEEE Symposium on Visual Languages and Human-Centric Computing. p. 1–8, 2005.

SCRATCH. Disponível em: <<http://scratch.mit.edu/>>. Acessado em: Outubro de 2016.

SHAYO, C., GUTHRIE, R., & IGBARIA, M. Exploring the measurement of end user computing success. Journal of End User Computing, 11(1), 5-14, 1999.

SHERMAN, Rick. How to evaluate the features of data integration products. Disponível em: <<http://searchdatamanagement.techtarget.com/feature/How-to-evaluate-the-features-of-a-data-integration-product>>. Acesso em: 24 out. 2016.

SIEWIOREK, D.P; GRAY, J. High-availability computer systems. Computer IEEE, Los Alamitos, CA, v. 24, n. 9, p. 39-48, set. 1991. Disponível em: <<http://ieeexplore.ieee.org/document/84898/>>. Acesso em: 21 nov. 2016.

SPAHN, Michael; DÖRNER, Christian; WULF, Volker. End User Development: Approaches Towards a Flexible Software Design. In: ECIS. p. 303-314. 2008.

STANTON, Jeffrey M. et al. Analysis of end user security behaviors. Computers & Security, [s.l], v. 24, n. 2, p.124-133, mar. 2004.

STOLEE, K. T.; ELBAUM, S.; SARMA, A. End-User Programmers and their Communities: An Artifact-based Analysis. International Symposium on Empirical Software Engineering and Measurement. p. 147–156, 2011.

STOLEE, K. T.; ELBAUM, S.; SARMA, A. Discovering how end-user programmers and their communities use public repositories: A study on Yahoo! Pipes. Information and Software Technology, Elsevier B.V. p. 1289–1303, Outubro 2012.

SUTCLIFFE, A. & MEHANDJIEV, N. End-user development: introduction. Communications of the ACM, p. 31-32, 2004.

TRIBUNAL DE CONTAS DA UNIÃO. Riscos e Controles para o Planejamento da Contratação, Brasília, DF, v. 1.0, 2012. Disponível em: <<http://portal.tcu.gov.br/lumis/portal/file/fileDownload.jsp?fileId=8A8182A24D6E86A4014D72AC82195464>>. Acesso em: 12 dez. 2016.

UNION INVESTMENT. Union Investment Improves Operational Efficiency and Competitiveness by Developing Intuitive Web Application to Manage 960 Investment Funds. Disponível em: <<http://www.oracle.com/us/corporate/customers/customersearch/union-investment-1-apex-2660622.html?ssSourceSiteId=otnen>>. Acesso em: 13 dez. 2016.

USABILITY TESTING (Singapore). Conducting Heuristic Evaluation. Disponível em: <<http://usabilitytesting.sg/lesson-8-conducting-heuristic-evaluation/>>. Acesso em: 24 out. 2016.

USERSCRIPTS. Disponível em: <<http://userscripts.org/>>. Acessado em: Outubro de 2016.

ZIMMERMANN, S.; RENTROP, C; FELDEN, C. Managing Shadow IT Instances – A Method to Control Autonomous IT Solutions in the Business Departments. 20th Americas Conference on Information Systems (AMCIS), 2014.

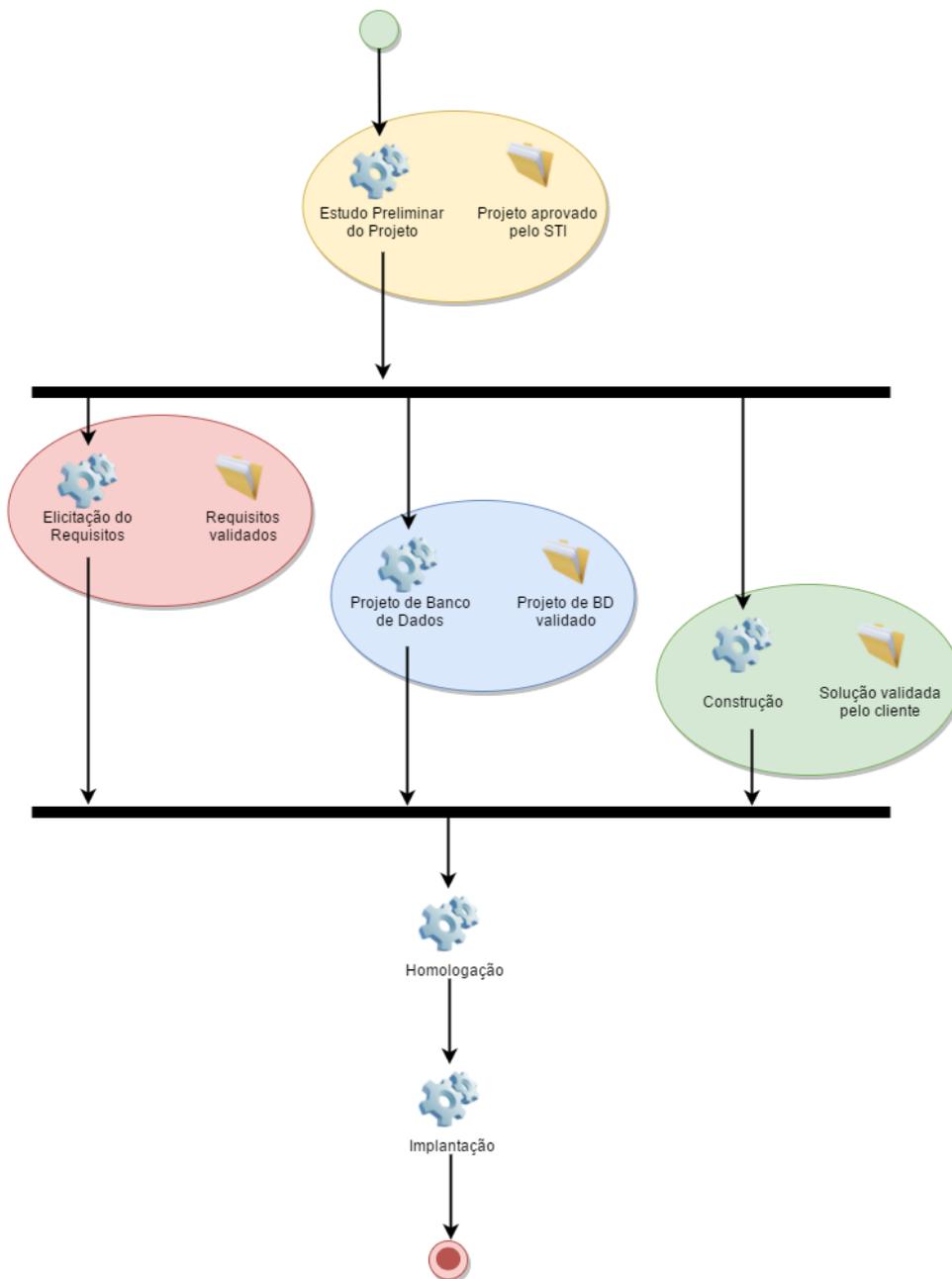
## 7 APÊNDICE A: PROCEDIMENTOS, PAPÉIS E RESPONSABILIDADES DO EUD NO ÓRGÃO PÚBLICO A

Na Tabela 10 apresentam-se as atividades realizadas no processo de desenvolvimento descentralizado de soluções de software do Órgão Público A.

*Tabela 10 - Atividades e Papéis Responsáveis*

Área de Negócio	Área de TI
<b>Identifica necessidade / problema de negócio e formaliza demanda</b>	Analisa a necessidade / problema de negócio (viabilidade e soluções)
<b>Aloca equipe de desenvolvimento (Servidor ou Estagiários)</b>	Seleciona e capacita a equipe
<b>Define prioridade, escopo e prazos</b>	Orienta e acompanha o projeto de desenvolvimento (escopo, risco e prazos)
<b>Desenvolve e documenta o sistema</b>	Orienta e acompanha o desenvolvimento de soluções (requisitos, modelagem e interface)
<b>Treina os usuários</b>	Provê acesso a dados corporativos por meio de <i>views</i> e efetua testes e homologação
<b>Implanta e mantém o sistema</b>	Presta consultoria por meio de reuniões agendadas (métodos e técnicas)
<b>Atualiza o portfólio de sistemas</b>	Provê infraestrutura (banco de dados, ferramenta RAD, ferramentas de colaboração)

O ciclo de vida do processo de desenvolvimento descentralizado do Órgão Público A é apresentado na Figura 5.



*Figura 5 - Ciclo de vida do processo de desenvolvimento descentralizado do Órgão Público A*

## 8 APÊNDICE B: CONCEITOS DE EUD E EUC

Na Tabela 11 apresentam-se diferentes conceitos de EUD presentes na literatura.

*Tabela 11 - Conceitos do EUD (end user development) no decorrer do tempo, baseado em Barker (2007) e Barker (2011)*

EUD ( <i>end user development</i> )	
Autores	Conceitos
<b>Sutcliffe and Mehandjiev (2004)</b>	É sobre assumir o controle, não somente personalizar aplicações computacionais (EUC) e escrever programas, mas de projetar novas aplicações computacionais sem nunca ver o código de programa subjacente.
<b>Chaffey &amp; Wood (2005, p.559)</b>	“O termo real de end user development se referiria, no entanto, a um não-especialista em TI que criasse suas próprias aplicações para suportar o próprio trabalho”
<b>McGill (2004)</b>	Utiliza-se do conceito de Chaffey & Wood (2005, p.559), e o leva um pouco mais longe ao identificar que o desenvolvimento não poderia apoiar somente o trabalho do <i>end user</i> que realizou o desenvolvimento, mas, também, o trabalho de outros <i>end users</i> no departamento ou na organização.
<b>Jawahar and Elango (2001)</b>	Parte significativa do sistema organizacional de desenvolvimento com a capacidade para desenvolver pequenas aplicações formando que fazem parte dos requisitos de trabalho para muitas posições.
<b>Lieberman et al (2006)</b>	Conjunto de métodos, técnicas e ferramentas que permitem aos usuários de sistemas de software, que estão atuando como desenvolvedores de software não-profissionais, em algum momento criar, modificar ou estender um artefato de software.

Na Tabela 12 apresentam-se os diversos conceitos de EUC no decorrer do tempo.

Tabela 12 - Conceitos do EUC (end user computing), baseado em Barker (2007) e Barker (2011)

EUC - (End user computing)	
Autores	Conceitos
<b>Davis e Olson (1985)</b>	Capacidade dos usuários controlarem diretamente suas próprias aplicações e necessidades computacionais.
<b>Amoroso (1988, p. 50)</b>	Abordagem de desenvolvimento alternativa, em que o usuário pode evitar complexidades tradicionais de desenvolvimento, atrasos de tempo e problemas de comunicação.
<b>Brancheau and Brown (1993, p. 439)</b>	Adoção e o uso da tecnologia da informação pelo pessoal de fora dos departamentos de sistema da informação para o desenvolvimento de aplicações de software, a fim de auxiliarem em tarefas organizacionais.
<b>Rainer and Harrison (1993, p. 1188)</b>	Uso direto e individual do computador, abrangendo todas as atividades relacionadas ao PC requeridas ou necessárias para efetuar o trabalho que estiver realizando.
<b>Garavan and McCracken (1993)</b>	Uso administrativo e profissional do poder computacional em comparação com tarefas administrativas que usam o mesmo hardware de computador.
<b>Chan and Storey (1996, p119)</b>	Uso autônomo da tecnologia da informação por parte dos trabalhadores do conhecimento fora do departamento de SI.
<b>Martin et al. (2005)</b>	Atividade que envolve a implementação de todas as aplicações para diferentes níveis, tais como aplicações de apoio, aplicações pessoais e sistemas organizacionais.
<b>Downey and Bartczak (2005, p4)</b>	Uso e/ou o desenvolvimento de tecnologia computacional e aplicações de software por end users para solucionar problemas organizacionais e apoiar na tomada de decisões.
<b>Barker (2011)</b>	Uso da tecnologia computacional e/ou aplicações de software, junto com o aprimoramento, modificação e/ou desenvolvimento de sistemas de informação por end-users para o uso individual, departamental ou organizacional.

Amoroso (1988) aponta que o EUC já existia em várias organizações anos antes de 1988, de modo que na época dos mainframes, usuários já desenvolviam aplicações em períodos de baixo processamento empresarial. Esse fato reforça a ideia apresentada por Barker (2007), onde mostra o EUC como uma questão teórica que ganhou destaque no início dos anos de 1980 com a introdução de computadores pessoais.

Apresenta-se como um modelo objetivo e fornecedor de um framework a fim de gerir e controlar aplicações computacionais de *end-users* de forma holística. Essa ideia é explicada por Amoroso (1988) ao utilizar-se das pesquisas de Ives, Hamilton e Davis (1980): “Dado um ambiente com suporte e gerenciamento para o EUC (Ambiente Organizacional), o usuário (Ambiente do Usuário) é capaz de utilizar ferramentas de software e hardware para criar aplicações (Ambiente de Desenvolvimento)”. Atualmente, essa abordagem tem se tornado abundante em quase todo negócio, devido a diminuição dos custos dos computadores, a introdução de aplicações de software "*easy-to-use*" no mercado e a maior distribuição de *Software* livre (BARKER, 2007).