NATAL, 2018

Realização: SBC

Organização: UFRN

# anais 2018

## XXXVIII CONGRESSO DA SOCIEDADE BRASILEIRA DE COMPUTAÇÃO

### 3º WASHES – WORKSHOP SOBRE ASPECTOS SOCIAIS, HUMANOS E ECONÔMICOS DE SOFTWARE

CENTRO DE CONVENÇÕES | NATAL•RN | 22 A 26 DE JULHO DE 2018

**#COMPUTAÇÃOESUSTENTABILIDADE**



CSBC 2018

# anais 2018

**XXXVIII CONGRESSO DA SOCIEDADE BRASILEIRA DE COMPUTAÇÃO**
CENTRO DE CONVENÇÕES | NATAL•RN | 22 A 26 DE JULHO DE 2018
**#COMPUTAÇÃOESUSTENTABILIDADE**


**Coordenador Geral**

Francisco Dantas de Medeiros Neto (UERN)

**Comissão Organizadora**

Bartira Paraguaçu Falcão Dantas Rocha (UERN)
Camila Araújo Sena (UERN)
Everton Ranielly de Sousa Cavalcante (UFRN)
Felipe Torres Leite (UFERSA)
Ilana Albuquerque (UERN)
Isaac de Lima Oliveira Filho (UERN)
Priscila Nogueira Krüger (UERN)

**Realização**

Sociedade Brasileira de Computação

**Organização**

Universidade do Estado do Rio Grande do Norte

# CSBC 2018
# XXXVIII Congresso da
# Sociedade Brasileira de Computação

## Apresentação

Estes anais registram os trabalhos apresentados durante o XXXVIII Congresso da Sociedade Brasileira de Computação (CSBC 2018), realizado em Natal-RN, de 22 a 26 de julho 2018. O evento teve como tema central a Computação e Sustentabilidade, pois se compreende que o avanço da computação e as questões ambientais devem caminhar lado-a-lado, tendo em vista que as técnicas computacionais necessitam ser usadas para possibilitar o desenvolvimento sustentável, e, desse modo, equilibrar as necessidades ambientais, econômicas e sociais.

Organizar o maior evento acadêmico de Computação da América Latina foi um privilégio e um desafio. Foi enriquecedor promover e incentivar a troca de experiências entre estudantes, professores, profissionais, pesquisadores e entusiastas da área de Computação e Informática de todo o Brasil. Ao mesmo foi desafiador termos que lidar, principalmente, com às dificuldades impostas pelo momento de crise que o nosso Brasil vem enfrentando. Uma crise que afeta diretamente nossas pesquisas e, consequentemente, o desenvolvimento e inovação do nosso amado Brasil.

Por meio de seus 25 eventos, o CSBC 2018 apresentou mais de 300 trabalhos, várias palestras e mesas-redondas. O Congresso ainda abrigou diversas reuniões, que incluem a reunião do Fórum de Pós-Graduação, a reunião do CNPq/CAPES, a reunião dos Secretários Regionais SBC, a reunião das Comissões Especiais e a reunião do Fórum IFIP/SBC.

O sucesso do CSBC 2018 só foi possível devido à dedicação e entusiasmo de muitas pessoas. Gostaríamos de agradecer aos coordenadores dos 25 eventos e aos autores pelo envio de seus trabalhos. Além disso, gostaríamos de expressar nossa gratidão ao Comitê Organizador, por sua grande ajuda em dar forma ao evento; e, em especial, à equipe da Sociedade Brasileira de Computação (SBC), por todo apoio.

Por fim, reconhecemos a importância do apoio financeiro da CAPES, do CNPq, do CGI.br, do Governo do Estado do Rio Grande do Norte, da Prefeitura Municipal do Natal, da Prefeitura Municipal de Parnamirim, da CABO Telecom, da ESIG Software e Consultoria, da DynaVideo e do SENAI.

Natal (RN), 26 de julho de 2018.

**Chico Dantas (UERN)**
Coordenador Geral do CSBC 2018

**Anais do CSBC 2018**

# 3º WASHES – WORKSHOP SOBRE ASPECTOS SOCIAIS, HUMANOS E ECONÔMICOS DE SOFTWARE

## Coordenação do Evento

- Davi Viana (UFMA)
- Igor Wiese (UTFPR-CM)

## Comitê de Programa

- Adriano Albuquerque, UNIFOR
- Alexandre L'Erario, UTFPR
- Aline Vasconcelos, IFF
- Anna Beatriz Marques, UFC
- Claudia Cappelli, UNIRIO
- Edson Oliveira, SEFAZ/AM
- Elisa Huzita, UEM
- Emilia Mendes, Blekinge Institute of Technology
- Fernando Figueira Filho, UFRN
- Gislaine Camila Leal, UEM
- Gleison Santos, UNIRIO
- Heitor Costa, UFLA
- Henrique Cukierman, COPPE/UFRJ
- Igor Steinmacher, UTFPR
- Inaldo Costa, ITA
- Ivan Machado, UFBA
- Jonice Oliveira, UFRJ
- José Jorge L. Dias Jr., UFPB
- José Maria David, UFJF
- Julio Leite, PUC-Rio
- Karla Fook, IFMA
- Luis Rivero, UFAM
- Mauricio Aniche, Delft University of Technology
- Natasha Valentim, UFPR
- Regina Braga, UFJF
- Rodrigo Santos, UNIRIO
- Valdemar Vicente Graciano Neto, UFG
- Vanessa Nunes, UnB

# SUMÁRIO

## Artigos Completos



## Artigos Curtos

# Abordagens Metodológicas da Experiência dos Usuários Cegos Aplicadas nas Interações Web em Dispositivos Móveis: Uma Revisão Sistemática da Literatura

**Tiago Nogueira[1], Letycia Duarte[2], Simone Los[2], Júllian Luz[2], Deller Ferreira[3]**

[1]Instituto Federal do Tocantins (IFTO)

[2]Instituto Federal de Mato Grosso (IFMT)

[3]Instituto de Informática
Universidade Federal de Goiás (UFG)

{tiago.nogueira}@ifto.edu.br

***Abstract.*** *With the exponential growth in the scientific interest on assessing the blind users experience in the web environment, it is essential to develop studies that correlate usability features lined up to the accessibility features. At the same time, with technological resources advances and access to information through different devices, there is a gap in the UX investigation in mobile devices, specifically, in the blind users experience. Therefore, this work proposes a systematic literature revision, identifying the main blind users experience methods applied in the mobile devices interaction. Thus, 805 scientific articles were identified through the literature, which approached subjects like usability, blind users experience and mobile devices. Through result extraction, sixteen different applied methods were identified in the blind users experience on mobile devices interaction. It is pointed out the applicability of methods supported by expert reviewers, the observation techniques, trial studies, validation and conformity verification with the Web Content Accessibility Guidelines.*

***Resumo.*** *Com o crescimento exponencial no interesse científico em avaliar a experiência do usuário cego em ambientes na Web, torna-se imperativo trabalhos que correlacionam atributos de usabilidade alinhados às características de acessibilidade. Ao mesmo tempo, com o avanço dos recursos tecnológicos e o acesso às informações por diferentes dispositivos, há uma lacuna nas investigações sobre Experiência do Usuário (UX) em dispositivos móveis, especificamente, na experiência do usuário cego. Assim, este trabalho propõe uma revisão sistemática da literatura, identificando os principais métodos da experiência do usuário cego aplicados nas interações com dispositivos móveis. Dessa forma, foram identificados por meio da literatura, 805 artigos científicos, os quais endereçavam assuntos sobre usabilidade, experiência do usuário cego e dispositivos móveis. Por meio da extração dos resultados, identificou-se dezesseis diferentes métodos aplicados na experiência do usuário cego, nas interações com dispositivos móveis. Destaca-se a aplicabilidade dos métodos apoiados por revisores especialistas, as técnicas de observações, estudos experimentais, validação e verificação de conformidade com as Diretrizes de Acessibilidade do Conteúdo na Web.*

## 1. Introdução

Nos últimos anos, houve um crescimento exponencial no interesse científico em avaliar os aspectos subjetivos na Experiência do Usuário (UX), contribuindo assim para o entendimento nas Interações Humano-Computador (IHC). Ao mesmo tempo, com o avanço dos recursos tecnológicos e o acesso às informações por meio dos dispositivos móveis, há uma lacuna nas investigações sobre UX, especificamente, na experiência do usuário cego em dispositivos móveis [Borg et al. 2015].

Neste sentido, a UX estabelece uma relação com a Usabilidade, por meio de métodos de avaliação de Usabilidade maduros [Law and Abrahão 2014]. No entanto, os métodos de mensuração da UX baseiam-se basicamente, nos métodos consolidados da satisfação dos usuários, da eficiência e eficácia das aplicações e dos mecanismos de Usabilidade [Law and Abrahão 2014].

Segundo [Lallemand et al. 2015], o termo Usabilidade é demasiadamente estreito, não representando uma visão abrangente das interações dos usuários. Assim, a UX abarca além dos conceitos fundamentais da Usabilidade, os aspectos subjetivos das interações, por exemplo, as emoções dos usuários durante as interações em aplicações móveis.

De acordo com [Nogueira et al. 2017], com a mudança conceitual de Usabilidade para a UX, profissionais de IHC encontram novos desafios na evolução do design de interação. Assim, por meio da avaliação dos aspectos subjetivos da UX, em aplicações web, é possível identificar características emocionais que podem colaborar para a construção de interfaces mais acessíveis. Essas emoções podem ser classificadas como UX positiva ou negativa, podendo ser mensuradas em qualquer plataforma, entre elas, em dispositivos móveis.

Para [Tuch et al. 2016], trabalhar com UX negativas torna-se valioso. A compreensão dos determinantes negativos informa aos projetistas sobre as possíveis armadilhas na UX em produtos ou aplicações. Além disso, com uma ampla gama de fenômenos psicológicos, existem evidências de que as experiências negativas têm um impacto mais forte nas pessoas do que as positivas. Em uma estimativa, cinco experiências positivas são necessárias para compensar uma negativa [Tuch et al. 2016].

Não obstante, com o aumento de pessoas com deficiências buscando acesso aos recursos disponibilizados na web, problemas de acessibilidade e de UX de cegos tendem a aumentar [Nogueira 2015].

Assim, este trabalho propõe-se à realização de uma revisão sistemática da literatura, objetivando a identificação dos principais métodos da UX de cegos aplicadas nas interações com dispositivos móveis, publicados nos últimos 5 anos. Para tal, na seção 2 é realizada uma revisão sistemática da literatura; na seção 3 são apresentados os resultados e discussões; e na seção 4 as conclusões.

## 2. Revisão Sistemática da Literatura

Para [Nogueira et al. 2017], nos últimos anos houve um aumento significativo no interesse científico na busca do entendimento da experiência do usuário, principalmente sob os aspectos da acessibilidade na web. Ao mesmo tempo, com o acesso a diferentes tipos de dispositivos por usuários, principalmente usuários com necessidades especiais, surgem

novos desafios para a comunidade da Interação Humano-Computador (IHC). Neste contexto, surge a seguinte indagação: Q01 – Quais são os principais métodos aplicados na Experiência do Usuário Cegos nas interações com dispositivos móveis?

Dessa forma, para responder à questão de pesquisa Q01, esta revisão sistemática da literatura utilizou-se a abordagem proposta por [Wohlin 2014]. Assim, a subseção 2.1 apresenta a seleção das fontes de pesquisa adotados por esta revisão, a construção do protocolo de pesquisa e os critérios de inclusão e exclusão. Na subseção 2.2 será apresentado a execução desta Revisão Sistemática da Literatura e na subseção 2.3 a extração dos dados.

## 2.1. Planejamento da Pesquisa

Objetivando a identificação de artigos científicos relevantes, os quais endereçam assuntos sobre a experiência dos usuários, usabilidade e dispositivos móveis, as seguintes bases de pesquisa foram adotadas: a) ACM Digital Library; b) Google Scholar; c) IEEE Digital Library; d) Science Direct; e) Springer Link.

Esta revisão sistemática buscou artigos científicos publicados entre janeiro de 2011 a dezembro de 2016, em periódicos e revistas internacionais. Dessa forma, com base na questão de pesquisa (Q01), foram extraídas as seguintes palavras-chaves:"*user experience*", "*usability*", "*blind user*", "*mobile*". Assim, construiu-se a *String* de Busca P01 - ("*blind*" AND "*mobile*" AND ("*user experience*" OR "*usability*") OR "wcag").

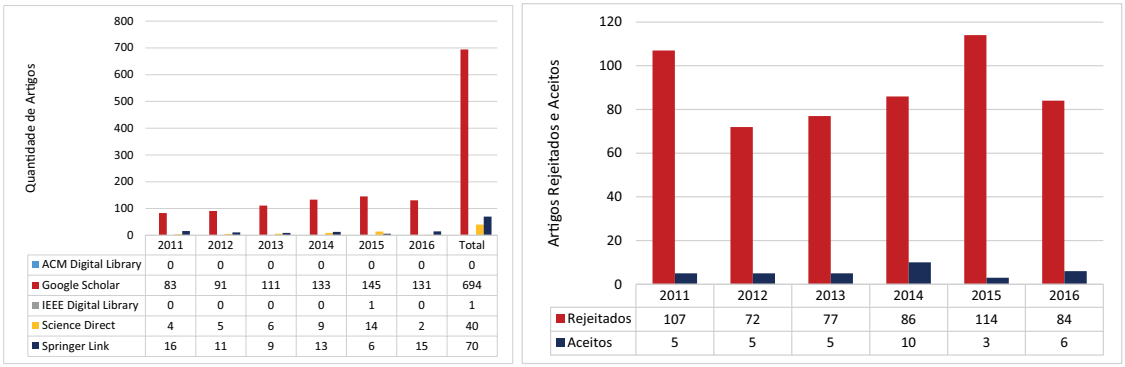
Com intuito de realizar testes preliminares de eficiência, ao aplicar a *String* de Busca no Google Scholar, foram identificados 25 artigos, os quais endereçavam assuntos sobre usabilidade e experiência do usuário. Desta forma a *String* de Busca mostrou-se suficiente para os objetivos desta pesquisa, trazendo-nos informações essenciais a respeito dos métodos mais propícios para a experiência do usuário cego em dispositivos móveis.

O processo de construção dos critérios de inclusão e exclusão deu-se com intuito de selecionar artigos os quais corroboram para resposta da questão de pesquisa Q01, isto é, para rejeição ou aceitação dos estudos. Os critérios de inclusão adotados nesta Revisão Sistemática da literatura foram: a) Trabalhos que utilizam-se de métodos para avaliação da UX de cegos aplicadas nas interações com dispositivos móveis; b) Trabalhos que investigam às experiências de usuários cegos, sob a perspectiva da usabilidade em dispositivos móveis; c) Trabalhos que definem métodos de inclusão a usuários cegos em dispositivos móveis.

Os critérios de exclusão adotados foram: a) Trabalhos que avaliam métodos de usuários que não sejam cegos; b) Trabalhos que apresentam métodos de usuários cegos, mas não em dispositivos móveis; c) Trabalhos que não se encaixam no período requisitado (5 anos); d) Trabalhos que não possuem resumo (*abstract*); e) Trabalhos que não avaliam métodos de usuários cegos em dispositivos móveis.

## 2.2. Execução da Pesquisa

Por meio da execução do protocolo de pesquisa (P01), foram identificados 805 artigos científicos, os quais endereçavam assuntos sobre usabilidade, experiência do usuário cego e dispositivos móveis. A Figura 1(a) apresenta o quantitativo de artigos científicos identificados por cada base.




| | 2011 | 2012 | 2013 | 2014 | 2015 | 2016 | Total |
|---|---|---|---|---|---|---|---|
| ACM Digital Library | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Google Scholar | 83 | 91 | 111 | 133 | 145 | 131 | 694 |
| IEEE Digital Library | 0 | 0 | 0 | 0 | 1 | 0 | 1 |
| Science Direct | 4 | 5 | 6 | 9 | 14 | 2 | 40 |
| Springer Link | 16 | 11 | 9 | 13 | 6 | 15 | 70 |

| | 2011 | 2012 | 2013 | 2014 | 2015 | 2016 |
|---|---|---|---|---|---|---|
| Rejeitados | 107 | 72 | 77 | 86 | 114 | 84 |
| Aceitos | 5 | 5 | 5 | 10 | 3 | 6 |

(a) Artigos científicos identificados por base/fonte

(b) Artigos científicos rejeitados/aceitos por período/ano

Figura 1. Quantitativo de artigos científicos identificados, rejeitados e aceitos

Foram identificados 694 artigos científicos na base Google Scholar, 1 artigo científico na base IEEE Digital Library, 40 artigos científicos na base Science Direct e 70 artigos científicos na base Springer Link. Observa-se que a quantidade de artigos identificados na base Google Scholar corresponde a 86% do total de artigos identificados por esta revisão.

Dos 805 artigos científicos identificados, 231 artigos foram classificados como "duplicados" (aproximadamente 28% da quantidade total dos artigos). Após a exclusão dos artigos duplicados, restaram 574 artigos, dos quais 34 foram "aceitos" e 540 foram "rejeitados". A Figura 1(b) apresenta o quantitativo de artigos que foram rejeitados/aceitos, por meio da aplicação dos critérios de exclusão e inclusão, nesta pesquisa.

Foram rejeitados 107 artigos científicos publicados em 2011, 72 artigos publicados em 2012, 77 artigos publicados em 2013, 86 artigos publicados em 2014, 114 artigos publicados em 2015 e 84 artigos publicados em 2016. Observou-se que, por meio dos critérios de exclusão, cerca de 67% do total dos artigos identificados foram rejeitados. Este fato ocorreu porque a grande maioria dos artigos científicos não endereçavam assuntos sobre a experiência do usuário cego em dispositivos móveis, apenas assuntos sobre a experiência do usuário vidente (que enxergam). Por meio da aplicação dos critérios de inclusão, foram aceitos 34 artigos. Observa-se que este número representa cerca de 4% do total de artigos identificados.

Por meio da aplicação dos critérios de inclusão, foram aceitos 5 artigos científicos publicados em 2011, 5 artigos publicados em 2012, 5 artigos publicados em 2013, 10 artigos publicados em 2014, 3 artigos publicados em 2015 e 6 artigos publicados em 2016.

### 2.3. Extração dos Dados

O processo de extração dos dados dos artigos que foram aceitos se deu pela leitura criteriosa por pares de revisores e pela supervisão de um pesquisador/especialista. Dessa forma, foram analisados 34 artigos científicos com intuito de identificar os principais métodos utilizados para a investigação da experiência do usuário cego em dispositivos móveis. A Tabela 1 apresenta os métodos identificados/extraídos por meio das análises/leituras dos artigos.

Tabela 1. Métodos de UX cegos em dispositivos móveis

| ID | Métodos X Autores |
|---|---|
| 1 | Heurísticas de Usabilidade [Mi et al. 2014]; |
| 2 | Revisão por especialistas [Pribeanu et al. 2014] [Neuschmid et al. 2012] [Leporini 2011] [Kerkmann and Lewandowski 2012] [Bose and Jürgensen 2014] [Loureiro et al. 2015] [Mi et al. 2014] [Buzzi et al. 2011] [Fakrudeen et al. 2014] [Milne et al. 2014] [Yesilada et al. 2011]; |
| 3 | Verificação/Avaliação de conformidade com a WCAG [Pribeanu et al. 2014] [Neuschmid et al. 2012] [Leporini 2011] [Babu and Singh 2013] [Murphy and Maycock 2013] [Bose and Jürgensen 2014]; |
| 4 | Inspeção por peritos em usabilidade [Leporini 2011] [Pribeanu et al. 2014] [Neuschmid et al. 2012]; |
| 5 | Estudos experimentais [Sultan et al. 2015] [Ferati et al. 2014] [Baumann 2012]; |
| 6 | Observação direta [Babu and Singh 2013] [Ferati et al. 2014] [Baumann 2012] [Kirinic et al. 2016] [Francis et al. 2013] [Kerkmann and Lewandowski 2012] [Bose and Jürgensen 2014] [Schoeberlein and Wang 2012] [Encelle et al. 2011] [Köhlmann and Lucke 2015] [Loureiro et al. 2015] [Milne et al. 2014] [Yesilada et al. 2011] [Ruiz et al. 2011] [Martínez and Pluke 2014]; |
| 7 | Verbalização do pensamento/ Verbal Think Aloud [Babu and Singh 2013]; |
| 8 | Questionários estruturados e abertos [Babu and Singh 2013] [Francis et al. 2013] [Schoeberlein and Wang 2012]; |
| 9 | Análise de dados de entrada (microfone/teclado) [Miao et al. 2016] [Façanha et al. 2014] [Ismirle et al. 2016]; |
| 10 | Frameworks de Usabilidade/Acessibilidade [Angkananon et al. 2014] [Francis et al. 2013]; |
| 11 | Testes com Usuários [Sierra et al. 2012] [Francis et al. 2013] [Bose and Jürgensen 2014] [Schoeberlein and Wang 2012] [Köhlmann and Lucke 2015] [Ismirle et al. 2016] [Loureiro et al. 2015] [Ismirle et al. 2016] [Façanha et al. 2014] [Pascual et al. 2014] [Sahasrabudhe and Singh 2016] [Yesilada et al. 2011]; |
| 12 | Modelos/Modelagem de Comportamentos/ Acessibilidade [Murphy and Maycock 2013] [Loiacono et al. 2013]; |
| 13 | Design centrado no Usuário/Aspectos Cognitivos [AlJarallah 2013] [Ismirle et al. 2016] [Al-Bassam et al. 2016] [Sahasrabudhe and Singh 2016]; |
| 14 | Prototipagem [Köhlmann and Lucke 2015] [Ismirle et al. 2016] [Al-Bassam et al. 2016] [Ismirle et al. 2016] [Fakrudeen et al. 2014]; |
| 15 | Estudo empírico/ Social empírico [Neuschmid et al. 2012] [Pribeanu et al. 2014]; |
| 16 | Método participativo [Mi et al. 2014]. |

Por meio da extração dos dados, foram identificados 16 diferentes métodos aplicados nos processos, de forma metodológica, nas experiências de usuários cegos em dispositivos móveis: revisão por especialistas; verificação/avaliação de conformidade com a WCAG; estudo empírico/social empírico; inspeção por peritos em usabilidade; estudos experimentais; observação direta; verbalização do pensamento/ *verbal think aloud*; questionários estruturados e abertos; análise de dados de entrada (microfone/teclado); *frameworks* de usabilidade/acessibilidade; testes com usuários; modelos/modelagem de comportamentos/ acessibilidade; design centrado no usuário/aspectos cognitivos; prototipagem; heurísticas de usabilidade; método participativo.

## 3. Resultados e Discussões

Nesta seção foram avaliados os métodos encontrados na Extração de Dados (subseção 2.3). Dentre os 16 métodos descobertos, destacou-se o método revisão por especialistas [Pribeanu et al. 2014] [Neuschmid et al. 2012] [Leporini 2011] [Kerkmann and Lewandowski 2012] [Bose and Jürgensen 2014] [Loureiro et al. 2015] [Mi et al. 2014] [Buzzi et al. 2011] [Fakrudeen et al. 2014] [Milne et al. 2014] [Yesilada et al. 2011]. Este método tem como objetivo identificar problemas de usabilidade para a pessoa com deficiência visual, visando tanto a acessibilidade quanto a usabilidade. O método verificação de conformidade de acordo com as diretrizes de acessibilidade (WCAG), possui o objetivo de disponibilizar uma visão geral das violações de acessibilidade e fornecer dicas úteis para revisores especialistas [Pribeanu et al. 2014] [Neuschmid et al. 2012] [Leporini 2011] [Babu and Singh 2013] [Murphy and Maycock 2013] [Bose and Jürgensen 2014].

O método social empírico, em estreita colaboração com o grupo-alvo e especialistas, conta com a aplicação de questionários estruturados [Neuschmid et al. 2012]. Os questionários são compostos por 5 seções com 55 perguntas. Neste método, é realizado um estudo empírico sobre os níveis de acessibilidade e a experiência do usuário cego, analisado do ponto de vista dos especialistas [Neuschmid et al. 2012] [Pribeanu et al. 2014].

Outro método identificado, por meio da extração dos dados dos artigos, inspeção por peritos em usabilidade, trata-se da avaliação da usabilidade por meio da aplicação dos princípios de estruturação do conteúdo, adequação do conteúdo e interatividade [Leporini 2011]. O princípio de estruturação do conteúdo está relacionado à partição lógica dos elementos de interface, por exemplo, o número de links da interface. Os princípios de adequação do conteúdo estão relacionados aos números de links apropriados, nomes apropriados para tabelas e imagens. A interatividade está relacionada à operabilidade da interface via teclados, sobre os mecanismos de mensagens e gestão dos dados [Leporini 2011].

Para [Sultan et al. 2015], por meio de estudos experimentais, pode-se avaliar aspectos da usabilidade em interfaces de dispositivos móveis por usuários cegos. Desta forma, avaliando a usabilidade de três telefones móveis, com aplicações *Android* distintas, é possível mensurar o grau de facilidade de uso, de aprendizado, da eficiência e compreensão dos usuários cegos durante as interações em diferentes aplicações [Sultan et al. 2015].

O método de observação direta tem como finalidade a realização da coleta de relatórios verbais de estudantes cegos

[Babu and Singh 2013] [Ferati et al. 2014] [Baumann 2012] [Kirinic et al. 2016] [Francis et al. 2013] [Kerkmann and Lewandowski 2012] [Bose and Jürgensen 2014] [Schoeberlein and Wang 2012] [Encelle et al. 2011] [Köhlmann and Lucke 2015] [Loureiro et al. 2015] [Milne et al. 2014] [Yesilada et al. 2011] [Ruiz et al. 2011] [Martínez and Pluke 2014]. Assim, os participantes trabalham em uma única tarefa e simultaneamente são verbalizados os pensamentos dos participantes durante as interações [Babu and Singh 2013].

Aplicando-se da observação direta, pode-se, por meio do método de avaliação da WCAG, usando apenas a análise de texto, interpretar e explicar os principais comportamentos dos usuários cegos nas interações em aplicações móveis. Assim, para obter melhores resultados, aplicam-se questionários estruturados e abertos [Babu and Singh 2013] [Francis et al. 2013] [Schoeberlein and Wang 2012].

Por meio da análise de dados de entrada do teclado, é possível identificar e avaliar a experiência do usuário nos processos de testes de usabilidade. Assim, pode-se utilizar dos componentes de entrada, teclado e microfone, para realizar observações do comportamento dos usuários durante as interações [Miao et al. 2016].

Observou-se, por meio das análises, a identificação de métodos que auxiliam aos desenvolvedores na criação de interfaces acessíveis, melhorando assim, a usabilidade das aplicações por usuários cegos [Angkananon et al. 2014]. Neste sentido, pode-se utilizar de *frameworks* para compreender os problemas e soluções enfrentadas por pessoas cegas [Angkananon et al. 2014] [Francis et al. 2013]. Assim, o Framework de Interatividade Aprimorada das Tecnologias (TEIF), consiste em 19 perguntas de múltipla escolha, o qual orienta ao desenvolvedor na construção de uma interface acessível e universal [Angkananon et al. 2014].

## 4. Conclusões

Esta revisão sistemática da literatura incidiu sobre as publicações científicas entre 2011 a 2016. Desta forma, foram identificados 805 artigos que endereçavam métodos da experiência do usuário cego aplicados nas interações com dispositivos móveis. Aplicando os critérios de inclusão e exclusão adotados nesta revisão, foram selecionados 34 artigos.

Percebeu-se, por meio da extração dos dados dos artigos, a identificação de 16 diferentes métodos aplicados na experiência do usuário cego, nas interações com dispositivos móveis. Assim, destacam-se a aplicabilidade dos métodos apoiados por revisores especialistas, empregando as técnicas de observações diretas, por meio de estudos experimentais, verificando e avaliando a conformidade com a WCAG. Observou-se também, as implementações de *frameworks*, objetivando a construção de interfaces acessíveis e com alto índice de usabilidade.

Não obstante, há um consenso entre os trabalhos, em processos de usabilidade e acessibilidade, ao mensurar a experiência do usuário cego, em utilizar-se da aplicabilidade de tarefas e questionários, por meio dos testes com usuários, avaliando o comportamento destes durante as interações. Neste sentido, percebeu-se o uso dos métodos de modelos comportamentais, os quais realizam a modelagem das interações dos usuários em determinadas aplicações.

Portanto, por meio da identificação dos principais métodos aplicados nos proces-

sos de mensuração da experiência do usuário cego em dispositivos móveis, os pesquisadores da área da Interação Humano-Computador (IHC) podem selecionar os métodos mais adequados, de acordo com seus objetivos, utilizando-se das melhores práticas da usabilidade e da acessibilidade na web.

## Referências

Al-Bassam, D., Alotaibi, H., Alotaibi, S., and Al-Khalifa, H. S. (2016). Easytrans: Accessible translation system for blind translators. In *International Conference on Computers Helping People with Special Needs*, pages 583–586. Springer.

AlJarallah, K. (2013). Cognitive user-centred design approach to improve accessibility for blind people during online interaction.

Angkananon, K., Wald, M., and Gilbert, L. (2014). Applying technology enhanced interaction framework to accessible mobile learning. *Procedia Computer Science*, 27:261–270.

Babu, R. and Singh, R. (2013). Enhancing learning management systems utility for blind students: A task-oriented, user-centered, multi-method evaluation technique. *Journal of Information Technology and Education: Research*, 12.

Baumann, K. (2012). Barrierefreiheit von facebook: Untersuchung mit hilfe des bitv-tests.

Borg, J., Lantz, A., and Gulliksen, J. (2015). Accessibility to electronic communication for people with cognitive disabilities: a systematic search and review of empirical evidence. *Universal Access in the Information Society*, 14(4):547–562.

Bose, R. and Jürgensen, H. (2014). Accessibility of e-commerce websites for vision-impaired persons. In *International Conference on Computers for Handicapped Persons*, pages 121–128. Springer.

Buzzi, M. C., Buzzi, M., Leporini, B., and Martusciello, L. (2011). Making visual maps accessible to the blind. In *International Conference on Universal Access in Human-Computer Interaction*, pages 271–280. Springer.

Encelle, B., Ollagnier-Beldame, M., Pouchot, S., and Prié, Y. (2011). Annotation-based video enrichment for blind people: A pilot study on the use of earcons and speech synthesis. In *The proceedings of the 13th international ACM SIGACCESS conference on Computers and accessibility*, pages 123–130. ACM.

Façanha, A. R., Viana, W., Pequeno, M. C., de Borba Campos, M., and Sánchez, J. (2014). Touchscreen mobile phones virtual keyboarding for people with visual disabilities. In *International Conference on Human-Computer Interaction*, pages 134–145. Springer.

Fakrudeen, M., Yousef, S., and Hussein, A. H. (2014). Analyzing app inventor for building usable touch screen courseware for blind users.

Ferati, M., Raufi, B., Kurti, A., and Vogel, B. (2014). Accessibility requirements for blind and visually impaired in a regional context: An exploratory study. In *Usability and Accessibility Focused Requirements Engineering (UsARE), 2014 IEEE 2nd International Workshop on*, pages 13–16. IEEE.

Francis, H., Al-Jumeily, D., and Lund, T. O. (2013). A framework to support e-commerce development for people with visual impairment. In *Developments in eSystems Engineering (DeSE), 2013 Sixth International Conference on*, pages 335–341. IEEE.

Ismirle, J., O Bara, I., Swierenga, S. J., and Jackson, J. E. (2016). Touchscreen voting interface design for persons with disabilities insights from usability evaluation of mobile voting prototype. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, volume 60, pages 780–784. SAGE Publications.

Kerkmann, F. and Lewandowski, D. (2012). Accessibility of web search engines: Towards a deeper understanding of barriers for people with disabilities. *Library Review*, 61(8/9):608–621.

Kirinic, V., Kozina, M., and Vidacek-Hains, V. (2016). Accessibility of information: International standards, recomendations and practices. *Economic and Social Development: Book of Proceedings*, page 47.

Köhlmann, W. and Lucke, U. (2015). Alternative concepts for accessible virtual classrooms for blind users. In *Advanced Learning Technologies (ICALT), 2015 IEEE 15th International Conference on*, pages 413–417. IEEE.

Lallemand, C., Gronier, G., and Koenig, V. (2015). User experience: A concept without consensus? exploring practitioners' perspectives through an international survey. *Computers in Human Behavior*, 43:35–48.

Law, E. L.-C. and Abrahão, S. (2014). Interplay between user experience (ux) evaluation and system development.

Leporini, B. (2011). Google news: how user-friendly is it for the blind? In *Proceedings of the 29th ACM international conference on Design of communication*, pages 241–248. ACM.

Loiacono, E. T., Djamasbi, S., and Kiryazov, T. (2013). Factors that affect visually impaired users' acceptance of audio and music websites. *International Journal of Human-Computer Studies*, 71(3):321–334.

Loureiro, J. R., Cagnin, M. I., and Paiva, D. M. B. (2015). Analysis of web accessibility in social networking services through blind users' perspective and an accessible prototype. In *International Conference on Computational Science and Its Applications*, pages 117–131. Springer.

Martínez, L. and Pluke, M. (2014). Mandate m 376: new software accessibility requirements. *Procedia Computer Science*, 27:271–280.

Mi, N., Cavuoto, L. A., Benson, K., Smith-Jackson, T., and Nussbaum, M. A. (2014). A heuristic checklist for an accessible smartphone interface design. *Universal access in the information society*, 13(4):351–365.

Miao, M., Pham, H. A., Friebe, J., and Weber, G. (2016). Contrasting usability evaluation methods with blind users. *Universal Access in the Information Society*, 15(1):63–76.

Milne, L. R., Bennett, C. L., and Ladner, R. E. (2014). The accessibility of mobile health sensors for blind users.

Murphy, M. P. and Maycock, K. (2013). Evaluating web accessibility for blind individuals.

Neuschmid, J., Hennig, S., Schrenk, M., Wasserburger, W., and Zobl, R. (2012). *Barrierefreiheit von online Stadtplänen–das Beispiel AccessibleMap*. na.

Nogueira, T. C., Ferreira, D. J., Carvalho, S. T., and Berreta, L. O. (2017). Evaluating responsive web design's impact on blind users. *IEEE MultiMedia*, 24(2):86–95.

Nogueira, T. d. C. (2015). Estudo comparativo da experiência de usuários cegos e videntes no design web responsivo e não responsivo.

Pascual, A., Ribera, M., Granollers, T., and Coiduras, J. L. (2014). Impact of accessibility barriers on the mood of blind, low-vision and sighted users. *Procedia Computer Science*, 27:431–440.

Pribeanu, C., Fogarassy-Neszly, P., and Pătru, A. (2014). Municipal web sites accessibility and usability for blind users: preliminary results from a pilot study. *Universal access in the information society*, 13(3):339–349.

Ruiz, B., Pajares, J. L., Utray, F., and Moreno, L. (2011). Design for all in multimedia guides for museums. *Computers in Human Behavior*, 27(4):1408–1415.

Sahasrabudhe, S. and Singh, R. (2016). Accessibility problems of blind mhealth users, a pilot study.

Schoeberlein, J. G. and Wang, Y. (2012). Accessible collaborative writing for persons who are blind: a usability study. In *Proceedings of the 14th international ACM SIGACCESS conference on Computers and accessibility*, pages 267–268. ACM.

Sierra, J., De Togores, J. S. R., and Selva, J. (2012). Designing mobile apps for visually impaired and blind users. In *The Fifth International Conference on Advances in Computer-Human Interactions*, pages 47–52. Citeseer.

Sultan, N., Siddiq, K., Rashid, T., and Farooque, M. (2015). Evaluation of smart phone applications accessibility for blind users. *Evaluation*, 127(3).

Tuch, A. N., Schaik, P. V., and Hornbæk, K. (2016). Leisure and work, good and bad: The role of activity domain and valence in modeling user experience. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 23(6):35.

Wohlin, C. (2014). Guidelines for snowballing in systematic literature studies and a replication in software engineering. In *Proceedings of the 18th international conference on evaluation and assessment in software engineering*, page 38. ACM.

Yesilada, Y., Brajnik, G., and Harper, S. (2011). Barriers common to mobile and disabled web users. *Interacting with Computers*, 23(5):525–542.

# Ethical design of social simulations

**Marco Almada, Romis Attux**

School of Electrical and Computer Engineering
University of Campinas (Unicamp)

marco.almada@usp.br, attux@dca.fee.unicamp.br

***Abstract.*** *Computer simulations of social phenomena are used in academic and non-academic contexts as tools for decision-aiding. By combining social scientific knowledge, computational power, and large data sets, those systems may be involved in decisions that affect people who are not even aware of the existence of a simulation. This paper reviews topics from the philosophy of technology and computer ethics literatures in order to identify salient ethical issues related to computer simulations and the steps on the software design cycle on which they can be addressed, offering a checklist that can guide how the ethical concerns of stakeholders are listened to and incorporated into the proposed simulation system.*

## 1. Introduction

Computer simulations of social phenomena play many roles at various aspects of modern society: they can be used in social-scientific studies [Elsenbroich and Gilbert 2014], as forms of entertainment [Koabel 2017], and within social credit systems [Yu *et al*. 2015] such as those currently in adoption in China [Hvistendahl 2018], among other uses. Since those applications play a growing role on how individuals and organizations make decisions, there are growing concerns about the transparency of the data and algorithms employed on simulations [e.g., Pasquale 2015], which may introduce or reinforce discriminatory patterns against socially vulnerable groups [O'Neil 2017; Eubanks 2018], issues that have been addressed both through technical approaches [ACM Public Policy Council 2017] and direct legal regulation [Ohm and Reid 2016].

To better understand how computer simulations can impact social life, this paper draws from a post-phenomenological philosophical framework [Verbeek 2005] to explain the different modes of interaction between computer simulations and individuals or groups. This analysis works as a start point for exploring ethical concerns such as biases and lack of accountability [O'Neil 2017], which then drive a targeted, non-systematic review of the simulation design literature. Finally, that review is then used as a basis for a checklist of measures to identify the relevant stakeholders and address their ethical concerns within the simulation development cycle.

This paper frames some of the current ethical issues related to computer simulations in terms of how they present themselves during the software development process. From this social and philosophical basis, simulation designers can then address ethical issues as additional requirements, to be treated according to best practices in software engineering. The proposed checklist identifies actionable points in which ethical concerns can be addressed, providing a starting ground that can be refined from the presented conceptual framework and the concrete demands of simulation projects.

## 2. Prior work

The object of the present discussion is how software developers can address ethical issues that arise when building computer simulations that describe or interact with social phenomena. Since those simulations have a wide range of applications, some domain-specific concerns will almost always be present, and an ethical framework for simulation must be adaptable enough to handle such cases. As a consequence, an approach that establish generic standards of ethical professional behavior will not suffice to cover entirely the ethical demands of simulation design, as Floridi and Sanders [2002] point out, and this section explores possible alternatives.

### 2.1. Computer simulations and social phenomena

Cioffi-Revilla [2014] presents a treatment of simulation-specific ethics, extending the *Truth-Beauty-Justice* framework proposed for assessment of formal models in social science. While beauty, defined in terms of formal aesthetics of modelling and programming, may have ethical consequences — for example, a minimalist simulation may be ethically desirable due to its reduced software and hardware demands, which in turn enable more people to run the simulation and reduce its ecological footprint —, most of the ethical concerns present in simulation design can be associated to truth and justice concerns. The *truth* factor in this ethical requirement can be associated with the epistemic requirements for using a computer simulation, that is, the conditions that must be met before one can trust that the simulation produces reliable results, such as accuracy; *justice*, in contrast, must be understood as a outward-looking target that relates the simulation and its modes of use to desirable social states.

The general topic of justice is object of a vast literature that could not be adequately surveyed within the scope of this paper; however, as Sen [2009] points out, a comparative approach can help ameliorate patently unjust situations even in the absence of a more general conception of a just society. In the context of computer ethics, one such approach can be seen at the end of Floridi and Sanders [2002]'s review of computer ethics, where the authors suggest that information-related notions such as information objects and entropy can be used to identify what are the relevant constraints in the environment in which simulations are deployed, allowing for a more apt description of the ethical demands related to a simulation in a given context.

An environmental approach becomes even more suitable when one takes into account that using computer simulations in ethical ways demands effort from a wide set of stakeholders: not only those responsible by programming the system itself, but also the model designers, those who actually use the models as part of their knowledge-acquisition or decision-making processes, among others. Even when they do not interact directly with the computer simulations, stakeholders can have their interactions with the world shaped by them; O'Neil [2017] cites the example of the 2016 US presidential elections, in which Clinton's decision not to focus on campaigning at the states of Michigan and Wisconsin, resulting in electoral losses at both states, was partially driven by the results of a simulation algorithm named Ada.

Verbeek [2005] presents a model to understand the ways in which technological artifacts can shape human interaction with other individuals and the world in general,

some of which which can be applied to computer simulations. A first mode of interaction between individuals and simulations is characterized by *hermeneutic* relations, in which simulations provide a representation of the world that must be interpreted by the user, as is the case when a hedge fund manager uses a Monte Carlo model as a proxy for deciding whether they should pursue or not a given investment. If, instead, an individual interacts with a simulation directly and not as a proxy — for example, when playing a computer simulation game —, the ensuing *alterity* relation emphasizes the (mostly) autonomous behavior of the simulational artifact.

Both forms of interaction, in Verbeek [2005]'s framework, are different from *background* relations, in which the presence of the artifact is barely noticed — or not at all — during normal operation: a bank customer's loan application may be conceded or denied based on simulation outputs and credit score values, but that person will hardly, if ever, be exposed do the actual simulation. As the growing availability of data allows the use of complex simulations in ever more important roles in business and government processes, individuals and communities will be subject to the economical, political, legal, and social consequences of decisions taken based on models that they do not know about and cannot control, making it more difficult to prevent unfair outcomes or to reverse them.

Lack of access to computer simulations becomes an even greater issue when one takes into account their *black box* nature: corporate practices, characteristics of algorithms and systems, and technical illiteracy among stakeholders [Burrell 2016] can prevent stakeholders from identifying or fixing unfair or otherwise inaccurate outputs that may arise from inadequate input data or other issues during simulation design and usage. This lack of understanding about computer simulations is compounded by what Pasquale [2015] termed a *black box society*: a complex social context in which interaction or even knowledge about simulations and related systems is mediated by specific rules, such as non-disclosure agreements and end-user licence agreements.

Due to those multiple sorts of opacity, non-user stakeholders rarely, if ever, have access to all the information needed for weighing the involved ethical factors, and even direct users might not be fully aware of the implications of adopting a given simulation model. Therefore, model and software designers must take an additional burden of understanding the actual consequences of placing computer simulations in relation with people, especially when those interactions happen in the background of the lives of individuals and groups, without their informed consent or knowledge.

### 2.2. Simulations as social systems

Discussions on the ethical aspects of computer simulation are nowadays usually happen within a more general debate on the use of computer-driven decision-making and *big data* [e.g., O'Neil 2017; Pasquale 2015; Ohm and Reid 2016; Eubanks 2018]. While those debates are usually framed around issues such as algorithmic fairness or algorithmic discrimination, they actually refer to *computer systems* which implement those algorithms and, by establishing relations between individuals and their environment such as those described in the previous subsection, affect not only their users but also third parties that might not even be aware of the system's existence.

Simulations can produce social effects by shaping modes of interpersonal interaction; they may thus lead to ethical violations if their design or use somehow conflicts with existing ethical duties, such as the respect for the privacy of its users. Those ethical hazards will depend on the specific nature of the relation mediated by a given simulation in a given context: for instance, simulations can lead to unethical results in a hermeneutical role if they fail to take into account relevant modeling aspects, while the background effect of simulation-aided decisions may force third parties into avoidable risks or unduly constrict their choices.

An example of ethical hazards induced by computer simulations can be seen from the risk models used by financial institutions prior to the 2008 economic crisis [Mackenzie and Spears 2014]. Those models, heavily reliant on the results of Monte Carlo methods and other simulations, guided investment decisions on banks; their normal operation brought significant profits, but failure to take into account possible modes of failure misled not only the direct users of those simulations — who then provided inaccurate advice to their internal and external customers — but also to the homeowners who held subprime mortgages and, ultimately, to the global economy.

O'Neil [2017] describes algorithms such as the aforementioned financial simulations or the ones used in prisoner parole evaluation as *weapons of math destruction*: computer systems that negatively impact the lives of millions of people without giving voice to the interests of those harmed. Those systems, by O'Neil [2017], are marked by three traits: they *damage* people's lives — by depriving them of options, as is the case on biased parole systems, or resources such as money —, operate at a *scale* that reaches thousands, if not more, of people — leading to feedback loops and emergent effects — and are *opaque*, in the senses discussed by Burrell [2016], preventing the parties harmed cannot understand what happens within the system in order to eliminate the negative outcomes.

As simulations are understood as intrinsically opaque models [Di Paolo *et al*. 2000], their design and use must be especially aware of how, deployed at scale in real contexts, simulation outputs might be used in ways that harms the rights of direct and background stakeholders. Still, as Doshi-Velez and Kortz [2017] aptly point out, the demands for algorithmic transparency are rarely absolute, as different contexts have different demands on what counts as a valid explanation of simulation outputs. Careful simulation design might, then, provide simulations with socially acceptable levels of result explainability, which can then be used to mitigate or avoid the damages they cause to the lives and rights of those directly or indirectly affected by the simulation.

### 2.3. Techniques for simulation design

Since computer simulations can shape, directly or indirectly, the perceptions and behavior of individuals far removed from the actual models, their design should take into account how the simulations can affect those people. A starting point for this can be drawn from the ACM Public Policy Council [2017]'s guidelines on algorithm transparency, which propose that computer models, algorithms, data, and decisions should be well-validated, well-understood, and auditable; furthermore, all stakeholders should be aware of possible harms coming from the resulting systems, and should be able to seek redress for any adverse effects.

Textbooks on the design of social simulations [such as Elsenbroich and Gilbert 2014; Gilbert and Troitzsch 2005; De Marchi 2005] usually bypass the potential ethical issues on simulation design and use. An exception, already discussed, is Cioffi-Revilla [2014]'s treatment of the *Truth, Beauty, and Justice (TBJ)* framework, which establishes criteria that can shape the requirements of the software engineering processes that build a given computer simulation. Operationalizing such values, however, requires that the simulation designers identify and address potential ethical concerns related to possible stakeholders, a task that can benefit from the existing software engineering literature.

Searching Google Scholar for all scholarly works between 2011 and May 2018 containing the terms "computer simulation" and "ethics" returns about 14,900 results. Most of that literature does not focus on design issues; in fact, adding the term "software engineering" to the search reduces the result set to 485 works. Even contemporary reviews of computing ethics [such as Stahl *et al.* 2016] do not emphasize simulation-specific issues. As a consequence, this paper is built around a directed discussion of core references cited by that literature rather than a systematic review of the works found by a refined version of the described search.

From a designer perspective, Ören *et al*. [2002] propose a code of ethics for designers of computer simulations; among the commandments established by this code, it is possible to identify some ethical issues — such as respect for intellectual property and due credit, avoiding harms to humans and the environment, and disclosure of system assumptions, limitations, and conditions of applicability — which address the social concerns presented so far in the text. As designed, though, the code shows the strengths and limitations of what Floridi and Sanders [2002] termed a "professional ethics" approach to computer ethics: they provide clear guides to action, but fail to address the rights and positions of other stakeholders, e.g. in background interactions.

As simulations of social domains can shape how their users and third parties interact with the world, their design should take into account the *persuasive* role that simulations take in such interactions. Davis [2009] provides an overview and discussion of persuasive software design approaches, and ultimately proposes a combination of two frameworks to ensure that any persuasion happens in ways compatible with the beliefs and values held by stakeholders. By using the *Value Sensitive Design* (VSD) framework, a systems designer can combine conceptual, theoretical, and empirical investigations to identify what values are relevant to stakeholders and should therefore be preserved to the largest extent possible. This approach can be complemented, as suggested in the same work, by *Participatory Design* (PD) methods that actively involve stakeholders in the various stages of the design cycle. Simulation designers can then obtain a fuller picture that will allow them to address in the concrete case design values such as the TBJ framework, result trustworthiness, and a taxonomy of ignorance [Williamson 2010] that describes how absence or distortions of knowledge and uncertainty or inaccuracy on the results can affect the decisions that are based on a given simulation.

## 3. Ethical constraints and the simulation development process

Computer simulations of social phenomena draw heavily on the mainstream software engineering literature. A representative example can be seen in Siegfried [2014]'s

proposal of a framework for agent-based models, which is divided in seven stages: a *preliminary* stage that identifies the needs of relevant stakeholders is followed by the construction of a logical *problem definition* that addresses the previously identified requirements. From this formulation, the simulation developers then produce a conceptual model of the target system through a *target system analysis* that is used for system *formalization* and *implementation*. Through an *experimentation* process, the ensuing computational system is adjusted so as to produce actionable insights through specialized *interpretation*. Each of those stages can be mapped to one or more of the traditional software engineering activities of specifying, designing and implementing, validating, and evolving a system[Sommerville 2011], but this domain-specific frame emphasizes how simulations are embedded into a social context of use.

Before identifying what are the needs and demands of each relevant stakeholder, one must identify *who* they are. Prior knowledge about the relevant phenomenon can be used to obtain a first approach to the set of affected people, but even so it will be rarely possible to include all of them into an actual software design process. Thus, some form of sampling might be necessary. A fair sample of the stakeholder population must capture a diverse range of positions, and a way to accomplish that is to oversample those populations that domain-specific knowledge identify as particularly vulnerable to ethical hazards from the simulation. While such oversampling can bias the relevant-stakeholder pool, it does so in a way that can help simulation developers to hear from perspectives that would otherwise be ignored; requirement analysis techniques can then be employed to identify unexpected side effects that would otherwise only be found after deploying the simulation system.

From the separate impressions of the relevant stakeholders, the simulation designers will then build a formal definition of the problem that must be solved by the resulting system. To do so, they must reconcile the ethical (and other) preferences and values identified in the previous step. In some cases, those ethical requirements can act as *side constraints* to the ultimate simulation goals: a simulation built to evaluate the installation of hypermarkets at a highly religious region can still provide useful information even if it must take into account the fact that a sizeable fraction of the stakeholders find it immoral to do business on specific dates. In other cases, the ethical requirements might make different demands to the system, but one such set of demands is more feasible than the other; here, the best problem formulation will be driven by the *consequences* of the possible sets of ethical constraints. Finally, it is possible that significant sets of stakeholders hold values that are mutually incompatible in their absolute forms, without any set being clearly preferable to another; any solution, then, will probably raise ethical issues for some stakeholders, and finding a working compromise is only possible by addressing on a way or another the concerns of all parts.

The VSD framework can be useful to find a solution that addresses such valorative concerns. Drawing from Davis [2009], this article proposes value scenarios — narratives that explore how people may use a system such as a simulation, identifying effects of the designed system on direct and indirect stakeholders over varying periods of time — as a tool to identify not only the critical values, but also relevant value clashes, such as *value sinks* strongly opposed by a subset of the relevant actors, and *value flows* that are not critical to the system itself but might be beneficial

due to their widespread acceptance. The results of those scenarios should then be empirically validated through the usual empirical software engineering approaches.

After identifying the relevant ethical issues relevant for simulation design, it is necessary to incorporate them into the logical model of the target phenomenon. That model can be either a closed mathematical system or a set of rules without a closed form, such as the interaction rules of an agent-based model [Elsenbroich and Gilbert 2014]. In this stage, the main concerns are related to model opacity, not only from the mathematical standpoint but also due to institutional constraints [Burrell 2016]; an answer to these forms of opacity is the explanation standard described by Doshi-Velez and Kortz [2017]: it should be possible to predict, given a set of inputs, how a specific factor can affect the output, based on the explanation requirements of stakeholders, identified for example through participatory design techniques [Davis 2009].

A valid logical model must then be implemented into an actual computer system. Since the software development processes involve specific technical expertise, including the relevant stakeholders in this stage might prove difficult, intensifying the opacity resulting form technical illiteracy. As a consequence, developers must take increased care on this stage, taking care to add as little opacity as possible to the logical models — or even introducing means for understanding the inner workings of the system, such as visualizations and result reproducibility —, while making sure that the models can be altered after construction so as to address issues identified during validation and also from the feedback obtained after deployment.

Procedures for ethical data collection are the subject of a wide literature that is not specific to the design of computer simulations, as discussed by Alvarez [2016]. Still, simulations themselves might generate significant amounts of data about their subjects, and that information must be stored in ways that do not expose the simulation targets and users to risk and that enable external validation, correction, and exclusion of information, as defined by laws such as the European GDPR. To address the issue of lack of feedback, that data must also be comparable to external outputs that allow simulation users and designers to identify prediction errors, which might prompt model redesign or a change in use cases.

Verification and validation processes can be a critical stage in ensuring that simulations behave in a proper way. In a technical sense, V&V processes should ensure that the resulting computer system can produce results within tolerable error margins and that it can handle any reasonably expected set of parameter and input values and operate at the required scales [Kaner and Swenson 2008]. Ethical hazards then can be treated as part of the validation criteria, to be evaluated in test cases with methods similar to those used for value extraction, and addressed at each stage of the cycle.

Validation by the team itself can be supplemented by external evaluations of the simulation design and use cycles, which can be conducted by regulatory organs, NGOs, or other trustworthy sources. An external perspective should allow the design team to address their own biases, but business, legal, or other demands might introduce secrecy requirements. This demand might be partially addressed through non-disclosure agreements and similar controls, and the combination of constant internal evaluation and external valuation will allow simulation designers to adjust simulation outputs to

reflect the actual results from the simulation-based interventions and adapt the constructed model to meet the ethical demands of stakeholder interactions.

## 4. A checklist for ethical simulation design

In this section, we synthesize the actionable insights from the previous section into a checklist for inspecting the development cycle of computer simulations. Checklists are an established tool for software inspection [Brykczynski 1999], which lend themselves easily to the evaluation of the entire simulation design; instead of finding software defects, this checklist will point sources of ethical hazards that should be inspected.

1. Is there an initial mapping of background relations involving the simulation?
2. Is there a clear sampling process for selecting background stakeholders to take into account?
3. Is there an unified model that reconciles the ethical demands of direct and background stakeholders?
4. Do the functional and nonfunctional requirements of the proposed system reflect the ethical demands from background stakeholders?
5. Is it possible to predict how variations in individual variables and parameters will affect the simulation output?
6. Is there a clear model of the possible sources of ignorance about the simulation results?
7. How does the computer implementation of the logical model add new sources of opacity to simulation?
8. Are there any interaction effects introduced by running the simulation with a large dataset and/or for a significant timespan?
9. Can the resulting simulation system be reasonably modified to handle new ethical constraints identified after its construction?
10. Are simulation results produced in a way that can be reproduced and compared with external data?
11. Can external stakeholders, directly or through representatives, propose changes to the finished model so as to address ethical concerns?

The presented checklist draws from good practices on checklist design that require non-trivial verification effort [Brykczynski 1999], but further work is required for empirical validation of the suggested practices. Still, the modular nature of the checklist means that it can easily be extended, shortened or altered based on the practical feasibility of this initial proposal.

## 5. Concluding remarks and further work

Whenever computer simulations are employed as decision-aiding tools in problems with social consequences, their use happens within ethical, legal and other forms of social constraints, which can either affect the means used by a simulation to achieve its objectives or even prevent that accomplishment at all. Still, the relationship between a social simulation and its running environment must not be seen as one-sided: through direct and indirect interactions with stakeholders, computer simulations can shape the ways in which people perceive their environment and act on it.

In such a context, simulation developers must be prepared to address the valid concerns of stakeholders that at first may seem to be significantly removed from the scenarios in which simulations are used. Respecting the needs and desires of those remote stakeholders may bring additional technical or social burdens to the simulation, such as the need for explaining the effect of each input variable. Yet, failing to do so might not only be unethical — by trampling the rights of other human beings without their consent or sometimes even their knowledge — but also illegal, resulting in blowback to systems designers and users.

This paper adopted a restrained approach, pointing out what kind of ethical demands are placed upon simulation developers by computer ethics and the philosophy of technology, and then showing how those requirements can be addressed within the usual software development processes. Further work is needed, especially for identifying relevant metrics for the explanation of simulation outputs and relevant stakeholder coverage, but the association between ethical concerns and established software engineering practices allow development teams to address the key concerns of system opacity, lack of feedback from real-world results, and lack of understanding of large-scale effects that have already caused unjust results in many spheres of contemporary societies.

## Acknowledgements

This work was partially funded by CNPq/Brazil (305621/2015-7). The authors would also like to thank three anonymous referees for their feedback.

## References

Alvarez, R. M. (2016). Introduction. In: Alvarez, R. M. (ed.) Computational Social Science. Cambridge University Press, 2016.

ACM US Public Policy Council. (2017). Statement on Algorithmic Transparency and Accountability.

Brykczynski, B. (1999) A survey of software inspection checklists. ACM SIGSOFT Software Engineering Notes, 24(1), 82.

Burrell, J. (2016). How the machine ‘thinks’: Understanding opacity in machine learning algorithms. Big Data & Society, 3(1), 2053951715622512.

Cioffi-Revilla, C. (2014). Introduction to computational social science. London and Heidelberg: Springer.

Davis, J. (2009). Design methods for ethical persuasive computing. In Proceedings of the 4th International Conference on Persuasive Technology. ACM.

De Marchi, S. (2005). Computational and Mathematical Modeling in the Social Sciences. Cambridge University Press.

Di Paolo, E., Noble, J., and Bullock, S. (2000) Simulation Models as Opaque Thought Experiments. In: Seventh International Conference on Artificial Life. Cambridge: MIT Press, 497-506.

Doshi-Velez, F. and Kortz, M. (2017). Accountability of AI Under the Law: The Role of Explanation. Berkman Klein Center Working Group on Explanation and the Law.

Elsenbroich, C. and Gilbert, N. (2014). Modelling Norms. Springer, 2014.

Eubanks, V. (2018) Automating Inequality: How High-Tech Tools Profile, Police, and Punish the Poor. St. Martin's Press.

Floridi, L., and Sanders, J. W. (2002). Mapping the foundationalist debate in computer ethics. Ethics and information Technology, 4(1), 1-9.

Gilbert, N. and Troitzsch, K. G. (2005). Simulation for the Social Scientist. Open University Press.

Hvistendahl, M. (2018) You are a number. Wired 26(1): 48-59.

Kaner, C. and Swenson, S. J. (2008). Good Enough V&V for Simulations: Some Possibly Helpful Thoughts from the Law & Ethics of Commercial Software. In Proc. Simulation Interoperability Workshop.

Koabel, G. (2017) Simulating the Ages of Man: Periodization in Civilization V and Europa Universalis IV. Loading..., *10*(17).

Mackenzie, D. and Spears, T. (2014) 'The Formula That Killed Wall Street': The Gaussian Copula and Modelling Practices in Investment Banking. Social Studies of Science, v. 44, 393–417.

Ohm, P. and Reid, B. (2016). "Regulating Software When Everything Has Software," 84 Geo. Wash. L. Rev. 1672-1702.

O'Neil, C. (2017). Weapons of math destruction: How big data increases inequality and threatens democracy. Broadway Books.

Ören, T. I., Elzas, M. S., Smit, I., and Birta, L. G. (2002). A code of professional ethics for simulationists. Proc. 2002 Summer Computer Simulation Conf. (San Diego CA).

Pasquale, F. (2015). The Black Box Society: The Secret Algorithms That Control Money and Information. Harvard University Press.

Sen, A. (2011). The Idea of Justice. Belknap Press.

Siegfried, R. (2014). Modeling and Simulation of Complex Systems: A Framework for Efficient Agent-Based Modeling and Simulation. Springer Viehweg.

Sommerville, I. (2011). Software Engineering, 9th ed. Addison-Wesley.

Stahl, B. C., Timmermans, J., and Mittelstadt, B. D. (2016). The ethics of computing: A survey of the computing-oriented literature. ACM Comput. Surv. 48, 4.

Verbeek, P.-P. (2005). What things do; philosophical reflections on technology, agency, and design. Pennsylvania University Press.

Williamson, T. J. (2010). Predicting building performance: the ethics of computer simulation. Building Research & Information, 38(4), 401-410.

Yu, L., Li, X., Tang, L., Zhang, Z., and Kou, G. (2015). Social credit: a comprehensive literature review. Financial Innovation, 1(1), 6.

# Análise da Percepção de Importância de Requisitos de Usabilidade no Desenvolvimento de um Sistema Web com *Scrum*

***Abstract.*** *The agile methodologies are used in software development projects, for their dynamic and efficient profile, to promote attending the customer satisfaction. However, in many cases, these methodologies seem to be used solely, without analyzing important aspects in the user-system relationship. Thus, this paper describes a case study about the perceived importance of usability in a Scrum team from a retail software company, and as a significant outcome of this work, it worths emphasizing that the team in question do not take into consideration the user study as essential practical task in order to address the usability requirements.*

***Resumo.*** *As metodologias ágeis são utilizadas nos projetos de desenvolvimento de softwares, por seu perfil dinâmico e eficiente, visando atender a satisfação dos clientes. Porém, em muitos casos, essas metodologias são utilizadas isoladamente, sem considerar aspectos importantes na relação usuário-sistema. Dessa forma, o artigo descreve um caso de estudo sobre a importância percebida da usabilidade em um time Scrum de uma empresa de softwares para o varejo e, como significante resultado deste trabalho, vale ressaltar que o time em questão não leva em consideração o estudo do usuário como prática essencial para atender requisitos de usabilidade do sistema.*

## 1. Introdução

A Engenharia de *Software* (ES) é caracterizada por padrões e métodos que norteiam o processo de desenvolvimento de *software*. Dentre os vários processos de desenvolvimento de *software* praticados pelo mercado de produção de sistemas, destaca-se a metodologia ágil. Segundo [Soares, 2004] essa escolha é baseada principalmente na versatilidade dos métodos ágeis, pois se adequam facilmente às mudanças, reduz a complexidade do planejamento, propõe um desenvolvimento rápido, aumentando o foco nas pessoas e não nos processos. Assim a equipe tende a ser mais colaborativa, visando entregar um sistema de alta performance.

Entende-se que é responsabilidade da equipe desenvolver um *software* usável e que atenda as expectativas dos clientes. Porém, apesar da interação com o cliente ser essencial no desenvolvimento ágil de *softwares*, pouco se fala da relação entre os métodos ágeis com aspectos que podem melhorar a qualidade do sistema para seu usuário final, como a usabilidade e o estudo do usuário.

Assim quando a aplicação de metodologias ágeis não leva em consideração aspectos de usabilidade, os sistemas, mesmo sendo planejados, desenvolvidos e entregues em concordância com os prazos, continuam correndo o risco de não serem aprovados por seus usuários. Seria natural então, em um cenário onde abordagens ágeis são interessantes alternativas para projetar um *software*, que pensássemos também em ações da área de usabilidade, pois essas disciplinas se complementam em alguns fatores que influenciam no valor agregado do produto final, como a garantia de satisfação e qualidade [Oliveira, 2016].

Apesar da relevância de uma pesquisa que discuta e analise a observância conjunta de elementos de Metodologia Ágil e Usabilidade, são raros os estudos de casos mostrando essa interação. [Silva, 2010] iniciou uma discussão sobre a relação de métodos ágeis com

Usabilidade, apresentando conceitos, as principais dificuldades em integrá-los e especificou as práticas ágeis mais adotadas (Sprints, Daily, User stories), técnicas de modelagens (protótipos, cenários de uso) e instrumentos (Designs Conceituais, Personas, testes rápidos);

[Cybis, 2009] implantou um estudo para avaliar a relação 'abordagem ágil *versus* engenharia de usabilidade' numa empresa de segurança da informação em Porto Alegre. O autor verificou as boas práticas que garantem o sucesso dessa interação, como a importância em fazer o *software* de acordo com protótipos; constatou também que as *Sprints* funcionam em vários ciclos do processo, como na análise, concepção e testes. Comprovou que testes rápidos alimentam as definições da próxima *Sprint*, o que garante a dinâmica ágil e produtiva. Porém, também verificou desafios a serem superados, como a concepção de interfaces em pares (programadores e designs). Cybis constatou que essa interação propõe uma maior análise dos processos de execução do sistema, mas também compromete a agilidade da entrega, necessitando assim de aperfeiçoamentos.

[Leite, 2013] aplicou um estudo de caso, avaliando duas equipes (uma ágil, denominada como equipe colaborativa e uma equipe tradicional) em uma empresa de desenvolvimento de *software*. As duas equipes possuíam experiências equivalentes em usabilidade. O estudo propôs uma análise das técnicas utilizadas pelas equipes e comprovou que a equipe colaborativa encontrou mais problemas de usabilidade no sistema; apresentou maior produtividade e mais compartilhamento de conhecimentos; demostrou maior alinhamento para integração em processos e ciclos interativos. O artigo comprovou no fim, a maior eficiência das equipes ágeis.

[Leite, 2013] corrobora que alguns desses poucos estudos mostraram que em ambientes onde são utilizados métodos ágeis ainda há muitos problemas a serem resolvidos quanto à integração com práticas de usabilidade. Diante da necessidade de maior explanação da temática, justifica-se o artigo aqui proposto, pois se verificou que essa discussão ainda é emergente, assim como observou-se, em um determinado ambiente profissional, a necessidade de analisar a importância que os membros da equipe fornecem à usabilidade no processo de construção de uma sistema.

Diante deste cenário, esse artigo tem como objetivo geral analisar a relação da Metodologia Ágil com aspectos de Usabilidade no processo de desenvolvimento de *softwares*. E como objetivos específicos (1) mapear como são tratados e definidos os aspectos de usabilidade no processo de planejamento e desenvolvimento de *softwares* construídos a partir de metodologias ágeis; e (2) verificar se a equipe que define o processo e desenvolve o sistema considera o usuário como parte essencial no processo de aceitação do produto.

Para tentar alcançar esses objetivos pretende-se entrar no universo de um projeto desenvolvido sob a vertente da metodologia ágil *Scrum*. A equipe estudada faz parte de uma empresa de desenvolvimento de *software* para o varejo. Os resultados obtidos e aqui apresentados podem contribuir como incentivo para exploração/criação de pesquisas mais abrangentes sobre a temática.

## 2. Métodos Ágeis

O conceito de Método Ágil tem como prioridade a "satisfação do cliente através de entregas antecipadas e contínuas de *software* de valor [Silva, 2010]". A partir desta premissa muitas organizações adotam processos ágeis no desenvolvimento de seus projetos com o intuito de oferecer o melhor produto no menor tempo.

Os princípios dos modelos ágeis deram origem ao Manifesto Ágil em 2001. O Manifesto surgiu estabelecendo as seguintes premissas:

- Indivíduos e interações mais que processos e ferramentas.
- *Software* em funcionamento mais que documentação abrangente.
- Colaboração com o cliente mais que negociação de contratos.
- Responder a mudanças mais que seguir um plano.

Entende-se assim, que as principais características da aplicação de métodos ágeis são: mais produtividade e menos custos, mais empenho e satisfação por parte dos colaboradores, entregas de *software* mais rápidas, maior qualidade no produto final e maior satisfação dos *stakeholders*[1].

Metodologias ágeis atuam no refinamento de metodologias interativas, tirando o foco do processo em si e dando mais ênfase para a contribuição das pessoas, assim a integração da equipe é essencial para entender como o desenvolvimento funciona e para definição da melhor abordagem para execução do trabalho**.** Para [Leite 2013] as reuniões diárias, prática comum aos modelos ágeis, são o principal momento onde as informações são trocadas, a equipe fica sabendo das atividades e dificuldades uns dos outros, assim, podem ajudar e aprender. Estas reuniões também são uma forma de incentivar o relacionamento entre a equipe.

Uma das metodologias ágeis mais difundidas da Engenharia de *Software* é a *Scrum.* Uma das características principais do método *Scrum* é se preocupar com a gestão do projeto e não com processos e controles. Assim com o *Scrum* é possível descobrir pontos falhos nos processos e como corrigi-los. O Scrum será detalhado a seguir.

## 2.1 Scrum

O *Scrum* é um método de gerenciamento que busca a flexibilidade dos prazos e resultados, utilizando times pequenos que colaboram entre si. Esse método não dispõe de técnicas ou processos específicos para o desenvolvimento do projeto, apenas estabelece conjunto de regras e práticas gerenciais que devem ser adotadas para o sucesso de um projeto [Carvalho & Melo, 2009]. Assim, fica a cargo da equipe definir a organização do processo, a execução das atividades, buscando sempre o sucesso na entrega do projeto.

*O Scrum* possui responsabilidades *(Product Owner, Scrum Master, Team),* atividades *(Sprint planning Daily Scrum, Sprint Review ) e* artefatos *(Product Backlog, Sprint Backlog).* Onde os responsáveis planejam a construção do produto. *As Sprints* são iterações do projeto, que podem durar de duas a quatro semanas, e *Backlog* consiste em uma lista das atividades a serem realizadas durante o projeto. No início de um *Sprint* é feita uma reunião de planejamento, na qual são selecionados determinados itens do *Backlog* que serão executados no período [Leite, 2013].

No *Scrum* é empregado o conceito de *time-boxes,* no qual cada atividade tem seu tempo estipulado, criando um controle na qualidade das funcionalidades e na produção. Essa autonomia faz com que a equipe se torne mais interativa e confiante, promovendo assim um desenvolvimento mais conciso.

A interação entre as pessoas que compõem a equipe *Scrum* é primordial, assim como a interação do usuário com o sistema. Para melhorar a relação usuário-sistema costuma-se aplicar aspectos de usabilidade.

[1] Público estratégico. Grupo ou pessoa que possui participação, investimento ou ações e que apresenta interesse em uma determinada empresa ou negócio. (BEZERRA, 2014).

# 3. Usabilidade

O termo usabilidade está relacionado com a facilidade de aprendizado e uso do sistema, bem como a satisfação do usuário em decorrência desse uso. Para [Nielsen e Loranger 2007, p.xvi] a usabilidade é um atributo de qualidade relacionado à facilidade do uso de algo. Mais especificamente, refere-se à rapidez com que os usuários podem aprender a usar alguma coisa, a eficiência deles ao usá-la, o quanto lembram daquilo, seu grau de propensão a erros e o quanto gostam de utilizá-la. Assim, [Siebra, 2011] afirma que especialmente no contexto de *software*, a interface deve ser configurada de forma a não exigir sempre dos usuários um novo aprendizado para associar comandos e ações, tornando a interação e uso mais "intuitivos", facilitando assim a ação e minimizando a probabilidade de erros.

Pensar na usabilidade de determinado sistema é sinônimo de pensar no usuário, em qualquer etapa do projeto, seja no início, meio ou no fim. Quando não se pensa na usabilidade desde o princípio do projeto pode-se perder tempo e recursos para, posteriormente, corrigir os eventuais erros e inconsistências que emergirão [Siebra, 2011]. Dessa maneira:

> a concepção de sistemas muitas vezes prioriza as exigências da informática antes de responder àquelas relacionadas ao usuário. Na maioria das vezes os profissionais de informática (projetistas) se empenham antes de tudo em definir as funções lógicas de um sistema sem de fato se preocuparam com as necessidades e habilidades físicas e cognitivas do usuário. [Silva, 2007 apud Ignácio; Carvalho, 2008].

No contexto do desenvolvimento ágil, a usabilidade ainda é tratada como algo complementar. Sendo apenas modificada quando há uma solicitação por parte do cliente, que em muitas vezes também não tem conhecimento sobre a importância da usabilidade no desempenho do trabalho do usuário final.

Existem várias estratégias para realizar a avaliação de usabilidade de um sistema, entre elas: os requisitos não funcionais de usabilidade [Pressman, 1992; Ferreira e Leite, 2003], Critérios Ergonômicos de [Scapin e Bastien 1993], Heurísticas de Usabilidade de [Nielsen 1994], e os princípios de Diálogo da ISO 9241:10, entre outros.

Para esta pesquisa abordaremos os requisitos não funcionais de usabilidade, por serem mais adequados para o contexto estudado.

## 3.1 Requisitos Não-funcionais (RNF)

Os RNFs, para qualquer tipo de sistema, estão relacionados a aspectos de *software*, *hardware* ou fatores externos, que determinem condições ou restrições ao comportamento do sistema pretendido [Sommerville, 1997, 2004] e [Pressman, 2004]. Entre estes requisitos é possível citar: desempenho, segurança, portabilidade, confiabilidade, manutenibilidade, acessibilidade, usabilidade, etc. Assim como os demais sistemas, os sistemas de *softwares* para varejo necessitam atender aos RNFs, os quais devem ser especificados com clareza.

De acordo com Pressman, os requisitos não funcionais de usabilidade da interface podem ser agrupados em duas categorias: os relacionados à entrada de dados e os relacionados à exibição da informação. [Pressman, 1992 *apud* Ferreira; Leite, 2003, p. 117]. Os requisitos relacionados à exibição da informação são:

**Quadro 1: Requisitos não funcionais de exibição da informação.**

| Requisito | Descrição |
|---|---|
| **Consistência** | Uniformidade dos comandos, funções, padrões. |

| | |
|---|---|
| ***Feedback*** | Resposta adequada do sistema ao usuário a toda e qualquer atividade. |
| **Níveis de habilidade e comportamento humano** | Garantia de acomodação de personalidades distintas, aplicando conhecimentos individuais. |
| **Percepção humana** | Preocupação com a qualidade de captação da informação. |
| **Uso de metáforas** | Facilitação da explicação e/ou compreensão de interfaces combinando conhecimento familiar e novos conceitos. |
| **Minimização de carga de memória** | Disponibilização de recursos que remetam ao cotidiano dos indivíduos |
| **Eficiência do diálogo, movimento e pensamento** | Simplicidade, clareza e objetividade na definição dos passos para realizar tarefas. |
| **Classificação funcional dos comandos** | Agrupamento das funcionalidades no menu do sistema de acordo com critérios conhecidos pelo usuário. |
| **Exibição exclusiva de informação relevante** | Eliminação de excesso informacional e de redundância de acessos a operações. |
| **Uso adequado de janelas** | Cautela na utilização de janelas simultâneas, a fim de evitar sobrecarga de informações. |
| **Manipulação direta** | Objetos visuais devem estar nítidos durante sua utilização. |
| **Uso de rótulos, abreviações e mensagens indicativas** | Clareza, objetividade e significância na elaboração destes elementos. |
| **Projeto independente da resolução do monitor** | Definição da aparência dos componentes utilizando proporcionalidade em relação ao espaço disponível. |

### Categoria Entrada de Dados

Muito tempo de trabalho do usuário é gasto com a escolha de comandos, digitação de dados e outros *inputs*. Um bom sistema deve otimizar ao máximo o tempo que o usuário gasta com essas tarefas. As diretrizes apresentadas a seguir tornam o sistema mais poderoso no que diz respeito à entrada de dados [Pressman, 2004].

**Quadro 2: Requisitos não funcionais de Entradas de Dados.**

| Requisito | Descrição |
|---|---|
| **Mecanismos de ajuda** | Disponibilização de informação de ajuda para toda ação de entrada. |
| **Prevenção de erros** | Desativação de caminhos inválidos para evitar erros do usuário. |
| **Tratamento de erros** | Recursos que permitam a correção de erros de forma ágil. |

## 4. Metodologia

A pesquisa é de natureza exploratória, pois se pretende levantar e analisar opiniões e práticas ainda desconhecidas sobre um grupo específico. Quanto aos objetivos, esta pesquisa tem caráter descritivo. Segundo [Gil 2008], as pesquisas descritivas possuem como objetivo a descrição das características de uma população, fenômeno ou de uma experiência. Quanto à fonte de dados pode-se considerar esta pesquisa como bibliográfica. Para [Marconi e Lakatos 2007], a pesquisa bibliográfica oferece meios para definir, resolver, não somente problemas já conhecidos, como também explorar novas áreas onde os problemas não se cristalizam suficientemente, tendo como objetivo permitir ao pesquisador um reforço na análise de suas pesquisas ou manipulação de suas informações. Considerando as abordagens de análise dos dados, esta pesquisa se caracteriza como quantitativa e qualitativa [Gil, 2008]. Com objetivo de derivar conclusões de forma clara e sistêmica, evidenciada a partir dos dados coletados.

Também foi utilizado na pesquisa o método indutivo de abordagem. Justifica-se a

utilização do método indutivo, pois a partir de dados coletados de um grupo de participantes, deseja-se chegar a uma teoria fundamentada, a partir de interpretações e abstrações, objetivando identificar e analisar fatores e aspectos [Felix, 2011].

Quanto ao procedimento adotado nesta pesquisa, destaca-se o estudo de caso holístico de caso único, onde um estudo de caso estuda um fenômeno dentro do seu contexto, principalmente quando os limites entre o fenômeno e o contexto não estão claramente definidos [Yin, 2005 *apud* Felix, 2011]; holístico porque a unidade de análise são as práticas aplicadas ao desenvolvimento do projeto pelo engenheiro de software (desenvolvedores, arquitetos de softwares, analista de testes), pois o objetivo da pesquisa é analisar os aspectos de atuação destes indivíduos ligados ao projeto; e único porque considera a organização estudada como caso representativo, que se deseja investigar a utilização das práticas de usabilidade com o método *Scrum.*

O universo da pesquisa baseia-se em uma equipe que desenvolve *softwares* direcionada pelo método *Scrum*. Esta equipe faz parte de uma empresa do segmento de *softwares* para o varejo, atuando há mais de 30 anos no mercado. Sua sede está situada em São Paulo (SP), mas possui filias espalhadas pelo Brasil, em cidades como Belo Horizonte (BH), Rio de Janeiro (RJ), Manaus (AM), Recife (PE), Franca (SP), Porto Alegre (RS), entre outras unidades, possuindo mais de 2.000 funcionários.

A equipe analisada faz parte das filiais Recife (PE) e Franca (SP) e contém 11 participantes, sendo 1 representante do cliente, 1 gerente do projeto, 1 analista de serviços, 2 analistas de suporte, 4 desenvolvedores e 2 analistas de testes.

Assim, os participantes foram contatados por e-mail e após aceitarem o convite foi enviado o link de endereço do questionário. Foi aplicado à equipe 12 questões envolvendo os RNFs de usabilidade relacionados ao desenvolvimento via metodologia *Scrum*. Essa medida visa levantar/analisar a percepção dos envolvidos em relação à interação do método ágil com os RNFs.

Para a aplicação do questionário foi utilizada a ferramenta aberta *Google Forms,* com o objetivo de melhor estruturar as respostas obtidas. Foi dado o prazo de cinco dias (de 03/07/2017 a 07/07/2017) para o preenchimento do questionário. Após esta etapa, os dados foram compilados e apresentados em figuras geradas pela própria ferramenta. A análise dos dados foi feita de forma qualitativa, a fim de fundamentar os resultados obtidos e evidenciados a partir dos dados.

Vale salientar que o cenário estudado é restrito e os resultados não devem ser generalizados como práticas comuns às demais equipes *Scrum* espalhadas pelas diversas empresas de softwares existentes.

## 5. Resultados

A priori foi feita uma análise dos aspectos intrínsecos à equipe estudada. Assim, percebeu-se que o projeto apresenta um escopo definido, com etapas de planejamento, definição das atividades, delegação das tarefas e prazos a serem atendidos. Dentro desse contexto, a equipe trabalha de forma dinâmica com *sprints* quinzenais.

Ressalta-se que dos 11 integrantes da equipe, dois não responderam a pesquisa, inclusive o representante do cliente, que não respondeu ao *e-mail* de solicitação de participação na pesquisa e um analista de suporte, que estava de férias no período de vigência do questionário. Salienta-se também que nesse estudo não foi feito um levantamento sobre o nível de conhecimento dos participantes sobre usabilidade

## 5.1 Usabilidade aplicada à metodologia Scrum

Ao serem questionados na pergunta 1 se há aplicação do estudo de usuário no planejamento do projeto, a maioria, 87% informou que sim, que há levantamento de necessidades e opiniões dos clientes. Vale salientar que no cenário estudado o cliente não é o usuário final do sistema, mas sim os operadores de frente de caixa das lojas. Estes não são consultados quando algum projeto é criado. Então, pode-se observar e presumir que no contexto do estudo o cliente pode não ser o melhor ponto focal para, de forma contínua, obter informações sobre as reais necessidades de quem opera o sistema, comprometendo seu desempenho e eficiência.

Já a pergunta 2, refere-se aos RNFs de usabilidade relatados nas Especificações, assim 45% falaram que as especificações abordam questões de usabilidade, 44% disseram que às vezes abordam e 11% informaram que não abordam. Compreende-se assim, que apesar da necessidade de clareza que a documentação deve conter, ainda é falha a questão da especificação dos RNFs nos projetos desenvolvidos na empresa. Provavelmente esse fato está ligado à cultura implantada, de que esses aspectos são considerados intrínsecos ao profissional que desenvolve o *software*. Normalmente o especificado refere-se à estrutura do código e ao retorno obtido quando determinadas funções são acionadas. Porém, em muitas situações, a não contemplação dos RNFs nas especificações compromete, consideravelmente, a utilização do sistema.

Essa suposição pode ser confirmada com os dados levantados na pergunta 3. Pois quando questionados se as instruções de usabilidade especificadas são suficientes para o entendimento e utilização do sistema por parte dos usuários, obtivemos o seguinte resultado: 43% afirmaram que às vezes as instruções de usabilidade são suficientes, 29% acreditam que as instruções não são suficientes e 28% acreditam que as instruções atendem completamente as necessidades dos usuários.

Entende-se, assim, uma divergência na opinião dos entrevistados, pois mesmo acreditando que as especificações contemplam os RNFs de usabilidade, os entrevistados alegaram também que nem sempre o que é especificado, é de fato entendido pelo usuário. O reconhecimento dessas dificuldades por parte do time faz-se entender que em muitas ocasiões quem está desenvolvendo tem problemas para traduzir ao cliente como a ferramenta deve ser utilizada.

Outro ponto referente aos RNFs foi o tratamento de mensagens de erros. Na Pergunta 4 todos os entrevistados responderam que o sistema contempla mensagens de erros. Porém, ao serem questionados sobre a clareza dessas mensagens na pergunta 5, 63% acreditam que nem sempre as mensagens retornadas informam claramente aos usuários o motivo exato que causou o erro no sistema; já 25% acreditam que as mensagens exibidas informam adequadamente ao usuário sobre os erros ocorridos e 12% acreditam que as mensagens não traduzem o ocorrido. Esse fato corrobora possíveis problemas de interação entre usuário e *software*. Pois para ser considerado fácil de usar, o sistema deve ser intuitivo, o usuário deve se sentir familiarizado com termos e funções. A falta de interação entre usuário e sistema acarreta retrabalhos e solicitações de mudanças no sistema, como mostra a questão seguinte.

Já quando se refere a mudanças no projeto (Pergunta 6), 56% informaram que sempre há mudanças e outros 44% informaram que as mudanças ocorrem ocasionalmente. Porém quando questionados em que momento as solicitações de mudanças ocorrem (Pergunta 7), todos afirmaram que as mudanças ocorrem durante o desenvolvimento. E dentre os motivos mencionados para tais mudanças (Pergunta 8), 80% informaram que as mudanças derivam de fatores externos, como solicitações de clientes, dificuldades na compreensão do sistema, mudanças de *softwares/hardwares*; já

20% acreditam que as mudanças ocorrem pela falta de conhecimento/entendimento da equipe desenvolvedora. Esse cenário denota possíveis problemas de comunicação entre o solicitado pelo cliente, e o que é entendido pelo time que desenvolve o projeto. Seria interessante, que a equipe buscasse formas mais objetivas de levantar e identificar as reais necessidades dos clientes, sem espaço para dúvidas.

Ao serem questionados sobre a utilização de práticas de usabilidade adotadas individualmente ou pela equipe (Pergunta 9), 63% informaram que não adotam nenhuma prática de usabilidade e 37% afirmaram a adoção. A respeito das práticas adotadas, foi mencionado o seguinte: ***"A prática adotada é de paralelismo entre Certificação e Homologação do produto. Este fator é utilizado devido à complexidade das metodologias ágeis e busca da maior qualidade do produto tendo em vista que a carga de testes pode ser compartilhada entre a equipe."***; entende-se aqui que os teste são divididos entre a equipe de desenvolvimento e a equipe de testes, para que assim haja uma cobertura maior dos testes; já outro entrevistado respondeu: ***"Destaque de funções mais utilizadas no sistema pelo usuário, afim de facilitar e agilizar o uso do mesmo. A ideia principal ao utilizar esta prática foi de ajudar o usuário durante a sua navegação, evitando que ele se desgaste durante qualquer percurso no sistema, tornando-o mais agradável. Além disso, estudos acerca do design, cores, estilos de botões, foram práticas de usabilidade que também são levadas em consideração na equipe de projeto o qual participei".*** Entende-se aqui que a estratégia é testar e aperfeiçoar as funções mais utilizadas pelos usuários.

A utilização de práticas isoladas de usabilidade denuncia uma possível lacuna na descrição desses aspectos na documentação, ocasionando a interpretação subjetiva por parte dos desenvolvedores, ou seja, o desenvolvedor presume e decide quais aspectos podem melhorar a utilização e compreensão do usuário. O que certamente causa divergências de compreensão e opinião entre a equipe, onde para um individuo uma certa funcionalidade deve ser desenvolvida sob um determinado ponto de vista, para outro indivíduo a mesma funcionalidade deve apresentar outro comportamento.

Quando questionados (Pergunta 10) se a solicitação de *feedback* é uma prática da equipe após a entrega das *Sprints Backlogs*, 78% disseram que sim, pois consideram importante o retorno dado pelos clientes durante essa fase, assim podem aperfeiçoar até a entrega final; já 22% afirmaram que às vezes esse *feedback* é necessário, depende da complexidade do projeto ou da necessidade de aprovação do cliente a cada etapa desenvolvida. Já sobre a satisfação do cliente (Pergunta 11), ao receber o produto, 75% dos entrevistados afirmaram que os clientes sempre estão satisfeitos, já 25% informaram que às vezes os clientes ficam satisfeitos. Essas opiniões são da equipe e não a opinião do cliente em si. Essa insatisfação pode ser ocasionada por problemas de comunicação entre cliente e time, onde o time não entregou exatamente o que o cliente solicitou ou pela não adequação do usuário ao sistema.

Por fim foi questionado (Pergunta 12) se o time acredita que o estudo do usuário é considerado relevante no planejamento do projeto. 56% informaram que sim, 33% às vezes e 11% que não. Esse cenário mostra que é comum a equipe atuar como 'intérprete de usuários', ou seja, supõem que seus costumes enviesados contemplarão todas as expectativas dos usuários, dispensando estudos e levantamentos sobre suas necessidades.

Outro ponto identificado é que a equipe '*Scrum*' apresenta divergências de opiniões e atuações, assim o entendimento sobre as necessidades do usuário é falho. Verificou-se que em várias situações a dinâmica *Scrum* não é completamente utilizada, comprometendo o resultado final do projeto.

## 6. Considerações Finais

Mesmo com os visíveis benefícios ofertados pela metodologia ágil *Scrum* no desenvolvimento de *software*, percebeu-se que no projeto estudado sua aplicação acontece, predominantemente, de forma isolada, sem interação com os usuários ou técnicas que tornariam o processo de desenvolvimento do *software* mais completo. A usabilidade, por exemplo, possui aspectos essenciais no desenvolvimento de *software* orientado ao usuário, porém pouco disseminado na cultura da equipe. O estudo de caso aplicado mostrou que a usabilidade é empregada de forma parcial, baseando-se nas percepções e definições da equipe desenvolvedora.

Verificou-se também que é prática comum, aos envolvidos no planejamento e execução do projeto, exercerem a função de Design de Interface de Usuário, aplicando conceitos de usabilidade limitados a sua experiência profissional e acreditando que estas são suficientes, considerando assim o estudo do usuário dispensável.

Essa visão distorcida é causa recorrente das alterações e revisões sofridas pelo projeto, tendo funções excluídas/inseridas pela não adaptação de seus usuários. Porém, mesmo com a identificação de problemas de interação entre desenvolvedores-clientes-usuários, o time ágil ainda duvida da importância do levantamento de perfil e necessidades do público alvo. O valor da usabilidade não está internalizado no time.

Pretende-se ainda estender essa pesquisa, através de uma proposta de mestrado, a várias equipes ágeis, em diferentes contextos organizacionais. Para assim, se ter uma análise mais profunda do cenário *Scrum* nas empresas de softwares. Esta pesquisa buscou apenas conhecer um cenário especifico, necessitando de estudos mais amplos.

## 7. Referências

Bezerra, F. (2014). O que é um Stakeholder?. *Portal Administração – Tudo sobre administração.* http://www.portal-administracao.com/2014/07/stakeholders-significado-classificacao.html.

Carvalho, B.V. and Mello, C.H.P. (2009). Revisão, Análise e Classificação da Literatura sobre o Método de Desenvolvimento de Produtos Ágil Scrum. In *XII Simpósio de Administração da Produção, Logística e Operações Internacionais*. São Paulo, Brasil: SIMPOI.

Felix, A. de L. C. (2011). *Um Estudo de Caso Sobre Motivação em Integrantes de Equipes de Desenvolvimento de Software no Contexto de uma Organização Pública*. Dissertação de mestrado. UFPE, Universidade Federal de Pernambuco, Recife, Brasil.

Ferreira, S. B. L. and Leite, J. C. S. do P. (2003). Avaliação da usabilidade em sistemas de informação: o caso do sistema submarino. *Revista de Administração Contemporânea – RAC*, 7(2), 115-137. http://www.anpad.org.br/rac/vol_07/dwn/rac-v7-n2-sbf.pdf>,

Gil, Antonio Carlos. (2008). *Como elaborar projetos de pesquisa* (5th. ed.). São Paulo, Brasil: Atlas.

Leite, S. F. C**.** (2013). *Inspeção de Usabilidade Aplicada a Métodos Ágeis: Um estudo de caso*. Monografia de Bacharelado. UFL, Universidade Federal de Lavras, Lavras, Brasil.

Marconi, M. de A. and Lakatos, E. M. (2004). *Metodologia Científica*. (3th. ed.). São Paulo, Brasil: Atlas.

Nielsen, J. and Loranger, H. (2007). *Usabilidade da Web*. Rio de Janeiro, Brasil: Elsevier.

Oliveira, G. R. (2016). *Integração de Práticas de Engenharia de Usabilidade em uma Abordagem Ágil de Desenvolvimento de Software*. Trabalho de conclusão de curso. UFSC, Universidade Federal de Santa Catarina, Florianópolis, Brasil.

Pressman, R. S. (2004). *Software Engineering – A Practioner's Approach*. (6th. ed.). Columbus, USA: McGraw-Hill education.

Siebra, S. A. and Santana, J. F. and Silveira, D. S. (2011). Analisando as Questões de Usabilidade e Acessibilidade do Portal de Periódicos da Capes. *Encontro Nacional de Pesquisa em Ciência da Informação*. Brasília, Brasil.

Silva, T. S. and Silveira, M. S. (2010). Integrando Avaliação de Usabilidade e Métodos Ágeis. In *V Mostra de Pesquisa da Pós-Graduação.* Rio Grande do Sul, Brasil: PUCRS.

Soares, M. S. (2004). Comparação entre Metodologias Ágeis e Tradicionais para o Desenvolvimento de Software. Universidade Presidente Antônio Carlos. Minas Gerais, Brasil.

Sommerville, I. (2004). *Software Engineering* (7th. ed.). Harlow, England: Addison-Wesley.

Sommerville, I. and Sawer, P. (1997). *Requirements Engineering: A Good Practice Guide*. Nova Jersey, USA: John Wiley & Sons.

# How Much Does It Cost? A Simulation-Based Method for Cost Prediction in Systems-of-Systems Acquisition Processes

**Valdemar V. Graciano Neto[1], Flávio E. A. Horita[2], Rodrigo P. dos Santos[3], Davi Viana[4], Mohamad Kassab[5]**

[1]Universidade Federal de Goiás, Goiânia – GO – Brazil

[2]Universidade Federal do ABC, Santo André – SP – Brazil

[3]Universidade Federal do Estado do Rio de Janeiro, Rio de Janeiro – RJ – Brazil

[4]Universidade Federal do Maranhão, São Luiz – MA – Brazil

[5]Pennsylvania State University, Malvern – PA – United States

valdemarneto@inf.ufg.br, flavio.horita@ufabc.edu.br, rps@uniriotec.br

davi.viana@lsdi.ufma.br, muk36@psu.edu

***Abstract.*** *Software economics, acquisition, and pricing are important concerns, in particular for Systems-of-Systems (SoS). SoS are alliances of independent software-intensive systems combined to offer holistic functionalities as a result of the constituents interoperability. SoS engineering involves separately acquiring constituents and combining them to form the SoS. Despite the existence of cost prediction techniques at Systems Engineering practice, predicting SoS acquisition costs at design-time should include: 1) an analysis of the minimum set of constituents that offer a 'good enough' result, and 2) an analysis of the compatibility between the constituents to deliver the expected result. The main contribution of this paper is proposing a novel simulation-based method for cost prediction in constituents acquisition process, while considering the effectiveness of constituents combination to offer the intended functionalities, and predicting the lowest configuration, at design-time. We adopt a simulation model to predict, at design-time, the results that shall be yielded by the constituents during SoS operation. Preliminary results point out the success of our method to predict such costs while still supporting a selection of the best architectural configurations.*

## 1. Introduction

Economic aspects are a pungent concern for both software production and software acquisition [Boehm and Sullivan 2000]. Software economics involves, in particular, an activity known as *software costing*, which consists of establishing a price for selling a software product (under the software vendor perspective) and assessing whether the price is fair considering the functionalities provided (from the client point of view). Software costing is actually required in software acquisition processes. Public governments open public announcements of software acquisition and software companies compete for selling their products. Companies establish prices for their software products and offer them. If they match the specification requirements with the lowest price, the government acquire their

software. However, with the emergence of smart cities, software pricing and acquisition processes have faced some additional challenges. Such process have involved, besides other concerns, the acquisition of multiple systems (e.g., Flood Monitoring Systems and Smart traffic systems) to form what is nowadays known as Systems-of-Systems (SoS[1]).

SoS comprise many independent software-intensive systems, known as constituents, that are combined to offer complex functionalities that could not be individually offered by their constituents. Since SoS depend on the compatibility among its constituents to achieve a cohesive mission, the design of a SoS should involve a careful selection of the participating constituents that exhibit the desired capabilities [Burton et al. 2014] and that best results to contribute to the accomplishment of the pre-established missions [Silva et al. 2015]. However, several candidate constituents may offer similar functionalities and thus it is important to consider other distinguishing factors such as the cost and predict how they will influence in the SoS holistic performance.

Acquisition of systems to be part of a larger set of interoperable systems is not a new trend, it has occurred since the 1970s in the USA, especially in the military domain [Acker 1983]. Satellites, airplanes, missiles, and systems have been purchased to interoperate for a long time during the last decades. However, the constituents are often individually acquired without (i) a thourough investigation on the value delivered when integrated within a larger system, (ii) a guarantee of functional compatibility, (iii) thorough investigation on the architectural configurations required to optimize the overall results, and (iv) a determination of the amount of constituents effectively needed to solve a problem. Evaluating costs and benefits at SoS context can be a complex task, since during its execution, a SoS can assume several distinct architectural configurations, which present different results that can influence the number of constituents required to be acquired, and the arrangement that should be maintained during SoS operation. Decisions made in the software development processes, especially in software architecture, have economic implications on the cost perspective. Therefore, it is important to investigate this economic aspect.

The main contribution of this paper is a novel simulation-based method to predict the optimal cost for SoS constituents acquisition. The method comprises three main steps: (1) the analysis of diverse architectural configurations a SoS can assume at runtime; (2) the selection of architectural configurations that offer the best combination of cost and performance; and (3) the assignment of an acquisition cost for that SoS based on a list of prices for each constituent system. Preliminary results reveal that our method is able to select the best architectural configurations, besides supporting a prediction of costs on the involved constituents. Such advance is important with the imminence smart cities (a type of SoS) and the respective government acquisition processes for SoS conception that may take place soon.

The paper is structured as follows: Section 2 presents the foundations to understand our proposal. Section 3 details our method, while Section 4 shows results of a preliminary evaluation. Section 5 discusses our results. Finally, Section 6 draws conclusions and indicates future work.

[1]For sake of simplicity, along this text, this acronym will be interchangeably used to express both singular and plural forms.

## 2. Background

SoS comprise a set of operational and managerial independent systems combined to offer larger functionalities that could not be individually delivered by any of them [Maier 1998]. Such complex functionalities are materialized as intended emergent behaviors, which can be intentionally engineered to accomplish a pre-defined set of missions [Rodriguez and Nakagawa 2017]. Individual missions are realized by constituent systems themselves whereas global missions of an SoS are accomplished through emergent behaviors [Silva et al. 2015]. SoS fulfill global missions by (i) performing assigned activities (individual missions) through constituents capabilities, and (ii) interactions among constituent systems leading to emergent behaviors.

SoS holds software architectures. A single software architecture comprises the fundamental structure of a software system, which comprises software elements, relations among them, and the rationale, properties, and principles governing their design and evolution [ISO 2011, Bass et al. 2012]. In turn, a SoS software architecture involves its fundamental structure, which includes its constituents and connections among them, their properties as well as those of the surrounding environment [Nielsen et al. 2015]. SoS software architectures are highly dynamic, i.e., they continuously change in response to addition, substitution, and deletion of constituents [Cavalcante et al. 2015]. In SoS software architectures, an *architectural configuration* is the current state and organization of an arrangement of interoperable software-intensive systems at a given point of time, also known as *coalition* in SoS domain. During its operation, a SoS software architecture can assume many different architectural configurations due to its *dynamic architecture* property. Each architectural configuration yields specific values about performance, reliability, and effectiveness. Such values can be collected through simulations, which enable an architect to anticipate, at design-time, the structure and behavior of a SoS before being deployed [Graciano Neto et al. 2018]. Once the best configurations are achieved, i.e., those systems that exhibit the best results with the lowest cost (the lowest number of constituents) are found, a self-healing mechanism can be triggered to maintain that coalition along the rest of the SoS operation, unless it occurs an emerging need of changing such structure. Therefore, coalitions can be predicted at design-time through simulations, and deployed to work later. Hence, the cost of system acquisition can be calculated in function of the predicted set of necessary (and enough) constituents, besides a margin of replacement (such as 10% of extra constituents) in case of defects or need of substitution.

Acquiring constituents to form such SoS depend on a twofold analysis: (i) the selection of constituents that offer the required set of capabilities necessary to fulfill the pre-established missions, and (ii) an assessment of the coalitions that offer the best results. Such results are often based on quality attributes, such as performance, and the available budget. Hence, constituents acquisition inherently involves a cost-benefit trade-off analysis, i.e., a balance between the advantages offered by a product and its associated cost.

Performance is one of the most important quality attributes to be analyzed in an acquisition process. At SoS context, performance concerns the results yielded by the synergy between the constituents systems to achieve the desired overall system goals[2].

[2] `http://www.sebokwiki.org/wiki/Architecting_Approaches_for_Systems_of_Systems`

Despite performance definition is not consensual for SoS context [Santos et al. 2014], for the scope of this paper, performance is defined as the degree of effectiveness of a coalition to accomplish a mission. This is measured as a relation between the effective number of data transported across a SoS architecture without losses and the total number of data provided as stimuli for feeding a simulation.

### 2.1. Related Work

SoS acquisition processes are often based on capability-based planning approaches, i.e., an optimization procedure that searches for a good solution that balances the set of desired capabilities and potential coalitions [Burton et al. 2014]. TLCM (Through Life Capability Management) [Urwin et al. 2010] and CapDEM [Robbins et al. 2005] are examples of approaches that rely on capability-based planning for predicting acquisition cost. However, those processes do not address an anticipation of the results exhibited by those coalitions measured in terms of quality attributes.

Burton et al. (2012) adopt a Model-Driven Engineering (MDE) approach, which includes domain-specific modeling languages to automatically generate potential solutions to the acquisition problem [Burton et al. 2012]. They progressed towards visualization techniques for the proposed solutions, and trade-off analysis for acquisition [Burton et al. 2014]. However, there is no focus on the results yielded by those potential solutions, specially considering quality attributes.

A recent work has invested on simulations for predicting attributes of a SoS software architecture at design-time [Graciano Neto et al. 2018]. In this approach, the authors specify a SoS software architecture using SoSADL models[3] [Oquendo 2016b], and automatically generating simulation models documented in DEVS [Zeigler et al. 2012]. After the assessment of multiple coalitions, the best configuration is elected. In the next section we show how such approach has been exploited for prediction of SoS acquisition cost.

## 3. A Simulation-Based Method to Support Constituents Acquisition for Systems-of-Systems Engineering

Figure 1 depicts our method that aims to help us predicting the selection of the best architectural configurations.

For determining the cost of system acquistion, the method starts with a list of consituent systems that goes through the following steps:

**Step 1. SoS Architectural specification in SosADL.** In this first step, an architecture of SoS is generated using SoSADL models;
**Step 2. Model transformation execution.** Having SosADL models produced, these are used as input for a model transformation to automatically generate simulation models specified in DEVS (a discrete-event simulation formalism)[4];
**Step 3. Simulation execution.** DEVS models produced in Step 2 are executed using MS4ME platform. Such Eclipse-based environment enables (i) the visualization of messages exchanged between constituents during SoS execution, (ii) dynamic architecture,

[3]SosADL is an architectural description language specially created for SoS domain.

[4]Details about the model transformation and how SosADL and DEVS models are mapped between them are not the focus of this paper and can be found in [Graciano Neto et al. 2018].

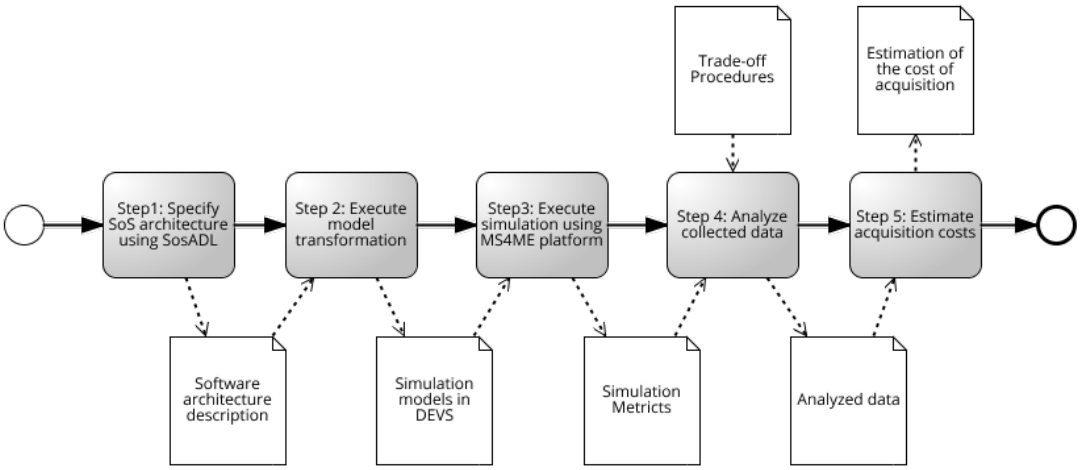



**Figure 1. A Simulation-Based Method to Support Constituents Acquisition for Systems-of-Systems Engineering**

and (iii) the measurement of pre-established metrics related to quality attributes;
**Step 4. Coalitions analysis.** After metrics collection, it is possible to analyze values delivered by coalitions through a trade-off procedure, supporting the decision of the coalition that offers the best combination between cost and benefits;
**Step 5. Cost estimation.** Once the constituents involved in that selected coalition are already defined, a table of prices can be used to estimate (with precision) the cost of acquisition for that set of constituents.

## 4. Evaluation

We conducted a pilot study to evaluate our method in supporting a precise prediction of acquisition cost in function of the selection of coalitions through simulations at design-time. A Flood Monitoring SoS (FMSoS) was adopted as an application scenario in this work. This SoS monitors rivers crossing urban areas, which pose great danger of floods in rainy seasons, potentially damaging property, lives, and possibly spreading diseases. FM-SoS notifies possible emergency situations to residents, businesses owners, pedestrians, and drivers located near the flooding area, and also to governmental entities and emergency systems. Its mission comprises ***the emission of flood alerts.*** FMSoS is composed of three different types of constituents:

1. **Smart sensors**, which are fixed embedded systems monitoring flood occurrences in urban areas, located on river edges;
2. **Gateways**, which gather data from constituents and share them with other systems;
3. **Crowdsourcing platforms**, which are mobile applications used by citizens for real-time communication of water level rising; danger level is a pre-defined value (between 1 and 6, 1 being no risk, and 6 being flood effectively occurring) that can be classified by a human user according to what he/she observes.

We provide a detailed overview on the execution of each step of our method for the application case (FMSoS).

**Step 1. SoS Architectural specification in SosADL.** In this step, SoSADL models were elaborated to register the FMSoS architecture. We modeled one of each type of constituent in SosADL, and replicated them.

**Step 2. Model transformation execution.** We run the model transformation and produced the simulation models. These were later deployed in MS4ME environment.

**Step 3. Simulation execution.** After deploying the DEVS models, we executed the simulation. It started with a configuration of constituent systems (i.e., four smart sensors, one gateway, and no crowdsourcing system). Along the execution, we exploited the dynamic architecture ability to include variations of the systems. This type of investigation enabled us to study the SoS behavior evolution based on its structural changes.

**Step 4. Coalitions analysis.** For this context, we compared coalitions considering the number of constituents of a coalition and their effectiveness to transport the data used to feed them. This analysis was done because the triggering of an emergent behavior is only possible as a result of the data exchange between constituents. Hence, the more successful a SoS architecture is to correctly transport data, the more effective it is to complete a given mission through its correspondent emergent behavior. Figure 2 plots the results of our analysis, taking into account (i) the percentage of the data fed to sensors that were correctly transmitted along the SoS architecture until the gateways considering the variation in the number of constituents, and (ii) the percentage of flood alerts that were triggered. It was observed that data loss increased with the number of sensors, reducing both the reliability of data transmission and triggered alerts (Point 2). This loss was alleviated by increasing the number of gateways, which increased the numbers of transmission rate and triggered alerts. When the architecture configuration had 40 constituents (Point 3), i.e., 30 sensors and 10 gateways (without considering mediators), the number of crowdsourcing platforms was increased as well. However, despite the expectation, increasing the number of crowdsourcing platforms neither increase the transmission rate, nor the number of alerts triggered because of the bottleneck of the gateways. Results improved again when the number of crowdsourcing platforms was fixed at 20 (Point 4), and the number of gateways was increased to 20 (Point 5), with 30 sensors, 20 gateways, and 20 crowdsourcing platforms (70 constituents, without considering mediators). It was also possible to observe that the rate of alerts correctly triggered was close to the rate of data effectively transmitted[5].

Good results occurred when FMSoS has many constituents, but these results are not better than when FMSoS has only five constituents. Hence, unless there is a situation in which a geographic area to be covered is too large, using a small number of constituents can achieve similar results as using a large number, at least for this domain, these configurations defined, and these types of constituents.

**Step 5. Cost estimation.** After selecting the coalition depicted in Point 1 of Figure 2, the cost estimation procedure was performed. We estimated prices for each type of constituent used in FMSoS in monetary units. As crowdsourcing systems are owned by population, only smart sensors and gateways were considered[6]. XBee technology is

[5]Results replicated from a previous study [Graciano Neto et al. 2018].

[6]SoS also depend on a type of element called mediator, which enables the forwarding of data between constituents, such as ZigBee wireless links. One mediator is required between each pair of constituents. We did not consider them in this study.

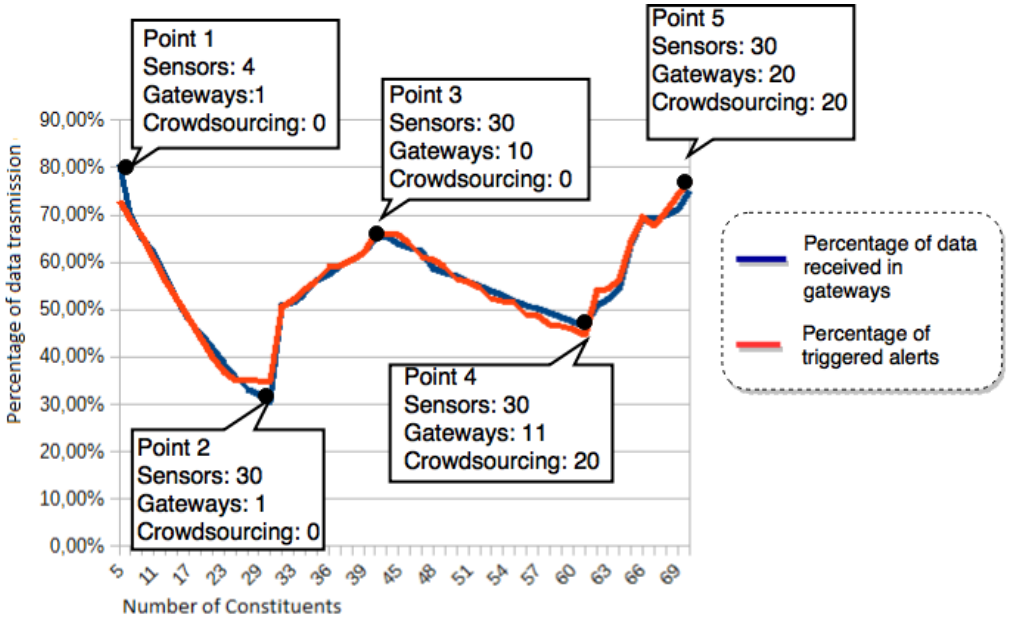



**Figure 2. Relation between percentage of data received in gateways and alerts triggered [Graciano Neto et al. 2018].**

one instance of smart sensor brand, whilst gateways can be materialized by an industrial computer linked to the Internet [Oquendo 2016a]. A smart flood sensor can be acquired from prices that varies from U$ 57.77 to U$ 79.99[7]. In turn, prices for gateways vary from U$ 28.95 to U$ 61.33[8].

Table 1 shows a comparison between realistic estimation of cost acquisition for smart sensors and gateways of four different brands. For this study, we assume that all of them deliver the same performance.

**Table 1. Comparison of prices between different constituent brands.**

| | *Brand A* | *Brand B* | *Brand C* | *Brand D* |
|---|---|---|---|---|
| **Smart Sensor** | $57.77 | $64.99 | $66.99 | $79.99 |
| | *Brand E* | *Brand F* | *Brand G* | *Brand H* |
| **Gateway** | $28.95 | $39.95 | $49.95 | $61.33 |
| Final Best Price for the selected coalition using cheapest prices among brands. | | | | $260.03 |

The lowest price found, as presented in Table 1, was $260,03. However, it is important to highlight that the acquisition costs could reach $2312,1 (Point 5 in Figure 2: 30 sensors and 20 gateways) if we have not performed an analysis that revealed a similar result between a small coalition and a large coalition. While this value is relatively high, our method demonstrates that the acquisition costs can be reduced in almost 90% ($2312,1 in relation to $260,03) in a SoS acquisition process due to a careful analysis of the results yielded by different coalitions. Moreover, in an acquisition process for military domain,

[7]https://goo.gl/9Lu5Ug
[8]https://goo.gl/cNxuFh

whose the order of magnitude of prices reaches millions of dollars, this type of procedure can be even more valuable.

## 5. Discussion

Our simulation-based method covers characteristics that are essential for SoS domain and that are not covered by other lines of studies on SOS, such as the dynamic architecture of the SoS, and analysis of emergent behaviors [Oquendo 2016b, Oquendo 2017]. Moreover, other proposals have been more focused on optimization problems to find, within the expected spectrum of constituent capabilities, the minimum set of constituents, without a thorough analysis of performance [Burton et al. 2014]. Our method analyzes the results delivered by the different coalitions according to a set of metrics (pre-defined in the context of the SoS architectural analysis approach), allowing a trade-off on metrics related to the quality and cost attributes of SoS software architecture.

This work also contribute to previous works on the role of architects of software ecosystems [Weinreich and Groher 2016, Amorim et al. 2017]. Within software ecosystems, the architect is resposible to define the optimal strategy of product because he knows the customers' needs and priorities. Results obtained in the evaluation of our method provide valuable metrics, as well as an organization of optimal arrangement of systems that would support a decision-making for software architects. In other words, they can make decisions considering not only customers' needs but also the lowest cost and best performance of systems.

We highlight the following threats to the validity of our conclusions: transformation correctness, human failure during estimation of prices, and choice of the best coalition. The same model transformation has already been used to make dozens of transformations between SosADL and DEVS models for two different domains: smart cities and space. Therefore, this threat is relieved by the number of studies that have already used such transformation. In addition, although formal proofs of its correctness have not been conducted, it generates correctly specified simulations every time. Such result is reliable because in the DEVS formalism a single erroneous instruction may make the simulation execution unfeasible, causing it to crash or even preventing its execution. From the point of view of human failure, there are some points in the process that are subject to failures, such as the observance and collection of metrics, as well as the choice among prices. A study with real data was performed on a small scale. However, results indicate the feasibility of reproducing it on a larger scale. Moreover, automation processes can be conducted to avoid human errors and enable the study of larger instances.

## 6. Final Remarks

Cost is a primary driver to decide whether to build a SoS or to create a new specialized system [Johnson 2015]. Moreover, cost is a relevant economic aspect of systems. Our simulation-based method enables to evaluate the performance of different arrangements of constituents and decide which constituents should be bought to form that SoS, offering the best performance and lowest cost.

After conducting a pilot study, we concluded that using a small number of constituents could achieve the same results as using a large number of constituents. Owing to such information, it is possible to anticipate which constituents are effectively necessary

to build a SoS, and predict the budget necessary to acquire them. The acquisition and construction of a SoS involves acquiring hardware in which software will be deployed with specific capabilities to realize the intended emergent behavior. In this paper, we exploited (i) the prediction of software architectures of a SoS at design-time, (ii) the prediction of different coalitions such architecture could assume at runtime, and (iii) the results that each one of such coalitions yield to suppport (iv) a prediction of cost of acquisition of the corresponding hardware necessary to support the existence of that SoS.

With the emergence of smart cities, software-intensive SoS will become highly open and dynamic. As such, dealing with acquisition cost comprises the prediction of their results at design-time, and the acquisition of the hardware in which the correspoding software will be deployed. Next steps of investigation include the prediction of software acquisition and software development under man-hour metric and function points prediction for development of software for constituents. Besides, our approach does not currently explore the diversity of constituents, i.e., we did not conduct a study in which products of different brands are benchmarked in order to choose not only the best coalitions, but also brands that offer lowest prices and better performance. This shall also be covered in forthcoming advances on this research. Other future work lines include (i) comparison among coalitions through the substitution of constituents that offer the same capability for better decision-making between different brands, (ii) adoption of co-simulation to accurately reproduce the scenarios required for other quality attributes such as security [Hachem et al. 2016], and (iii) establishment of a mechanism for automation of the cost estimation through the integration between the simulator, a mechanism for querying and comparing market prices, and some model-checker mechanism to automatically deliver the best coalition, without the need to manually collect and analyze data. We also consider that, for large volumes of data, we can apply search-based software engineering to support the selection of constituents from criteria related to technical and economic aspects of software. Nevertheless, we highlight the importance of the results achieved until now and the seminal nature of our solution for SoS domain.

## References

Acker, D. D. (1983). Defense systems acquisition review process: A history and evaluation. Technical report, Defense Systems Management Coll Fort Belvoir Va.

Amorim, S. S., McGregor, J. D., de Almeida, E. S., and von Flach G. Chavez, C. (2017). The architect's role in software ecosystems health. In *WASHES*, pages 1–4.

Bass, L., Clements, P., and Kazman, R. (2012). *Software Architecture in Practice*. Addison-Wesley Professional, Indianapolis, Indiana, USA, 3rd edition.

Boehm, B. W. and Sullivan, K. J. (2000). Software economics: A roadmap. ICSE '00, pages 319–343, New York, NY, USA. ACM.

Burton, F. R., Paige, R. F., Poulding, S., and Smith, S. (2014). System of systems acquisition trade-offs. *Procedia Computer Science*, 28:11–18.

Burton et al. (2012). Solving acquisition problems using model-driven engineering. In *ECMFA*, volume 7349, pages 428–443, Lyngby, Denmark. Springer.

Cavalcante, E., Batista, T. V., and Oquendo, F. (2015). Supporting dynamic software architectures: From architectural description to implementation. In *WICSA 2015*, pages 31–40, Montreal, Canada. IEEE.

Graciano Neto, V. V., Garcés, L., Guessi, M., Paes, C., Manzano, W., Oquendo, F., and Nakagawa, E. Y. (2018). ASAS: An approach to support simulation of smart systems. In *51th HICSS*, pages 5777–5786, Big Island, Hawaii, USA. IEEE.

Hachem, J. E., Pang, Z. Y., Chiprianov, V., Babar, A., and Aniorté, P. (2016). Model driven software security architecture of systems-of-systems. In *23rd Asia-Pacific Software Engineering Conference*, pages 89–96, Hamilton, New Zealand. IEEE.

ISO (2011). ISO/IEC/IEEE 42010:2011 - Systems and software engineering – Architecture description. *ISO*, pages 1–46.

Johnson, S. B. (2015). System health management. In Rainey, L. B. and Tolk, A., editors, *Modeling and Simulation Support for System of Systems Engineering Applications*, pages 131–144. Wiley, Hoboken, New Jersey, USA.

Maier, M. W. (1998). Architecting principles for systems-of-systems. *Systems Engineering*, 1(4):267–284.

Nielsen, C. B., Larsen, P. G., Fitzgerald, J., Woodcock, J., and Peleska, J. (2015). Systems of Systems Engineering: Basic Concepts, Model-Based Techniques, and Research Directions. *ACM Computing Surveys*, 48(2):18:1–18:41.

Oquendo, F. (2016a). Case study on formally describing the architecture of a software-intensive system-of-systems with sosadl. In *SMC 2016*, pages 2260–2266, Budapest, Hungary. IEEE.

Oquendo, F. (2016b). Formally Describing the Software Architecture of Systems-of-Systems with SosADL. In *11th Annual System of Systems Engineering (SOSE 2016)*, pages 1–6, Kongsberg, Norway. IEEE.

Oquendo, F. (2017). Architecturally describing the emergent behavior of software-intensive system-of-systems with SosADL. In *12th SoSE*, pages 1–6, Waikoloa, USA. IEEE.

Robbins, W., Lam, S., and Lalancette, C. (2005). Towards a collaborative engineering environment to support capability engineering. In *Proceedings of the 2005 INCOSE International Symposium*, pages 211–221, Rochester, NY, USA.

Rodriguez, L. M. G. and Nakagawa, E. Y. (2017). A process to establish, model and validate missions of systems-of-systems in reference architectures. In *SAC 2017*, pages 1765–1772, Marrakech, Morocco. ACM.

Santos, D. S., Oliveira, B., , Guessi, M., Oquendo, F., Delamaro, M., and Nakagawa, E. Y. (2014). Towards the evaluation of system-of-systems software architectures. In *8th WDES*, pages 53 – 57, Maceió, Brazil. SBC.

Silva, E., Batista, T., and Cavalcante, E. (2015). A mission-oriented tool for system-of-systems modeling. In *3th SESoS*, pages 31–36, Florence, Italy. IEEE.

Urwin, E. N., Pilfold, S. A., and Henshaw, M. (2010). Through life capability management: benefits and behaviours. In *International Conference on Contemporary Ergonomics and Human Factors*, pages 153–162. CRC Press.

Weinreich, R. and Groher, I. (2016). The architect's role in practice: From decision maker to knowledge manager? *IEEE Software*, 33(6):63–69.

Zeigler, B. P., Sarjoughian, H. S., Duboz, R., and Souli, J.-C. (2012). *Guide to Modeling and Simulation of Systems of Systems*. Springer-Verlag, London, United Kingdom.

# BPEL4PEOPLE Anti-Patterns: Discovering Authorization Constraint Anti-Patterns in Web Services

**Henrique J. A. Holanda**[1], **Carla K. de M. Marques**[2], **Francisca Aparecida P. Pinto**[3], **Yann-Gaël Guéhéneuc**[4]

[1] Department of Computer, State University of Rio Grande do Norte
Mossoró, RN, Brazil, 59.610-210

{henriqueholanda@uern.com}

[2]Federal Institute of Rio Grande do Norte
Mossoró, RN, Brazil,

{carla.marques@ifrn.edu.br }

[3]Department of Computer, Federal Rural University of the Semi-Arid
and State University of Rio Grande do Norte
Mossoró, RN, Brazil, 59625-900

{aparecidapradop@gmail.com}

[4]Department of Computer Engineering, University of Montreal,
École Polytechnique de Montréal

{yann-gael.gueheneuc@polymtl.ca}

***Abstract.*** *Despite the abundance of analysis techniques to discover anti-patterns in BPEL, there is hardly any support for authorization constraint errors in web services orchestrated by BPEL4People. Most techniques simply abstract from people (human user interactions), while people dependencies can be the source of all kinds of errors. This paper focuses on the discovery authorization constraint anti-patterns in web services orchestrated by BPEL4People. We present an analysis approach that is expressed in terms of rule card, the well-known, stable, adaptable, and effective model-checking techniques can be used to discover authorization constraint errors. Moreover, our approach enables a seamless integration of control-flow and authorization constraint verification.*

## 1. Introduction

A BPEL4People is a software system that manages and executes operational processes involving people, applications, and/or information sources on the basis of process models. BPEL4People systems are driven by explicit process models, i.e., based on a process model, a system is configured that supports the modeled process where exist user interactions. In this paper, we primarily focus on the analysis of the models used to configure web services orchestrated by BPEL4People. However, our approach is also applicable to other software system that manages and executes operational processes involving people. In the last 15 years, many analysis techniques have been developed to analyze web services orchestrated by BPEL4People. Most analysis techniques focus

on verification, i.e., the discovery of design errors. Although many process representations have been used or proposed, most researchers are using Petri Nets as a basic model [H.J.A Holanda and Serra 2010]. The flow-oriented nature of BPEL4People processes makes the Petri Net formalism a natural candidate for the modeling and analysis of work flows.

Unfortunately, lion's share of attention has been devoted to control-flow while ignoring other perspectives such as data-flow and resource allocation. Analysis techniques typically check for errors such as deadlocks, live locks, etc. while abstracting from data and uses. Existing approaches typically suffer from the following two problems: i) they look at only one perspective in isolation (e.g., only control-flow); and ii) the types of errors they capture are usually not configurable and mainly driven by the verification algorithms themselves rather than by user requirements [Dumas et al. 2005].

To address some of the limitations of existing approaches, we propose a new analysis framework based on uses of "anti-patterns" expressed in terms of rule card. Assuming a rule card representation, we define anti-patterns related to the BPEL4People. The term "anti-patterns" was coined in 1995 by Andrew Koenig. He stated that "An anti-pattern is just like pattern, except that instead of solution it gives something that looks superficially like a solution but isn't one" [Moha et al. 2012]. The goal of anti-patterns is to formally describe repeated mistakes such that they can be recognized and repaired. In this paper, we use rule card to formalize our anti-patterns. This formalization can be used to discover the occurrence of such anti-patterns in BPEL4People. Although not elaborated on in this paper, the same techniques can be used to define correctness notions related to the control-flow and check these in an integral way.

An example of an anti-pattern is Dangling Inputs Group of Users (DIGU). This anti-pattern describes the situation where some group of users' needs to be attributed for one task, but either it has never been created. The remainder of this paper is organized as follows. Section 2 presents related work. Section 3 introduces BPEL and BPEL4People. Section 3.2 introduces anti-patter. Section 5 presents the specification of authorization Constrain anti-pattern in web services orchestrated with BPEL4People. Section 4 presents our proposed approach. Section 5.1 presents the experiments and results. Section 6 concludes the paper and gives suggestions of future works.

## 2. Related Work

In the past years, several catalogs to specify and detect anti-patterns of SOA [Moha et al. 2012], [Palma et al. 2014b], Rest [Palma et al. 2014a] and BPEL [Palma et al. 2013] services and languages have been proposed. In [Smith and Williams 2003] the authors detected and evaluated more new software antipatterns and in [Sinnig et al. 2005] the authors detected and evaluated patterns in model-based engineering. We can highlight the drawbacks of the current literature as follows:

- Anti-patterns and approaches to detect them were considered only for SOA [Palma et al. 2014b] , [Palma et al. 2014a] and [Palma et al. 2013] models;
- Extending BPEL Engines with BPEL4People [Holmes et al. 2008];
- A model checking approach to verify BPEL4WS workflows [Bianculli et al. 2007];

- Web Services Human Task (WS-HumanTask) [Ings et al. 2012];
- Access control and authorization constraints for WS-BPEL [Bertino et al. 2006];
- Verifying BPEL workflows under authorization constraints [Zha 2006];

Finally, there is no detection approach for detecting authorization constraint anti-patterns in web services orchestrated by BPEL4People anti-patterns until now, so we focus on those issues with a solution to propose a concrete approach for specifying and detecting authorization constraint anti-patterns in web services orchestrated by BPEL4People.

## 3. BPEL4People and Anti-Patterns

Web Services Business Processes Execution Language focuses on business processes that orchestrate Web service interactions. However, in general, business processes are comprised of a broad spectrum of activities that most often require the participation of people to perform tasks, review or approve steps and enter data — for example, a credit approval scenario that may require approval on certain transaction limits or activity levels. These human interactions are now addressed in the new specifications. Human user interactions are currently not covered by the WS-BPEL, which is primarily designed to support automated business processes based on WS. In practice, however, many business process scenarios require user interaction. So far, we've seen that user interaction in business processes can get quite complex. Although BPEL specification 1.1 (and the upcoming BPEL 2.0) doesn't specifically cover user interactions, BPEL is appropriate for human work flows. Work flow services that leverage the rich BPEL support for asynchronous services are created today. In this fashion, people and manual tasks become just another asynchronous service from the perspective of the orchestrating process and the BPEL processes stay 100% standard.

### 3.1. BPEL4People

We now see the next generation of work flow specifications emerging around BPEL with the objective of standardizing the explicit inclusion of human tasks in BPEL processes. This proposal is called BPEL4People and was originally put forth by IBM and SAP in July 2005. Other companies, such as Oracle, have also indicated that they intend to participate in and support this effort.

The most important extensions introduced in BPEL4People are people activities and people links. People activity is a new BPEL activity used to define user interactions; in other words, tasks that a user has to perform. For each people activity, the BPEL server must create work items and distribute them to users eligible to execute them. People activities can have input and output variables and can specify deadlines. To specify the implementation of people activities, BPEL4People introduced tasks. Tasks specify actions that users must perform. Tasks can have descriptions, priorities, deadlines, and other properties. To represent tasks to users, we need a client application that provides a user interface and interacts with tasks: it can query available tasks, claim and revoke them, and complete or fail them.

To associate people activities and the related tasks with users or groups of users, BPEL4People introduced people links. People links are somewhat similar to partner links; they associate users with one or more people activities. People links are usually associated with generic human roles, such as process initiator, process stakeholders, owners, and

administrators [Ings et al. 2012]. BPEL4People extends the capabilities of WS-BPEL to support a broad range of human interaction patterns, allowing for expanded modeling of business processes within the WS-BPEL language. BPEL4People is comprised of two specifications including: i) WS-BPEL Extension for People which layers features on top of WS-BPEL to describe human tasks as activities that may be incorporated as first-class components in WS-BPEL process definitions; ii) Web Services Human Task introduces the definition of stand-alone human tasks, including the properties, behavior and operations used to manipulate them. Capabilities provided by Web Services Human Task may be utilized by Web services-based applications beyond WS-BPEL processes.

### 3.1.1. Integrating Authorization Constraints

BPEL4People support features to exclude some users from performing a task because of some tasks they had done before or force some user to perform a sequence of tasks. We call such requirement as authorization constraint, as the term is widely used in access control literature. In this section we will use GSPN to express the authorization constraints to facilitate formal analysis. Two kinds of authorization constraints, namely "4-eyes principle" and "chained execution", are proposed in BPEL4People specification. The "4-eyes principle", also known as "separation of duty", is a common scenario in many application areas when a decision must be made by two or more people independently of one another, often for the security reasons, and "chained execution" refers a process fragment where a sequence of steps must be executed by one person.

### 3.1.2. Separation of duty

The separation of duty (SoD) is a well-known principle in authorization to prevent fraud or error by requiring that at least two individuals are involved in some specific work. SoD is also useful when two persons have to co-operate in a work but none of them should know all the details. The basic form of SoD states that two given distinct tasks t1 and t2 must be performed by different individuals. This can be defined as states that person p0 cannot perform both t1 and t2. We can define variations of this similarly, e.g., "task t1 and t2 must be performed by different roles". We can also define SoD constraint for a specific person, e.g., "person A cannot invoke both task t1 and t2".

### 3.1.3. Binding of duty

"Binding of duty" (BoD) is the dual of SoD, which states that some distinct tasks must be performed by one person. BoD is used to define the responsibility of a person, e.g.: It states that if p0 performs t1, then p0 must also perform t2, and vice versa. SoD and BoD may be combined to define more complex constraints.

## 3.2. Anti-Patterns

Changes resulting from the evolution of orchestrated with BPEL4People can degrade its design of web services and can often cause the appearance of poor solutions in the architecture: anti-patterns.

Patterns and anti-patterns exist to capture expertise, and to communicate knowledge. Anti-patterns are opposed to patterns which are good specifications (solutions) for recurring problems. An anti-pattern is a surface-level symptom that hints at the presence of a deeper, more serious problem. Anti-patterns could hinder [Palma et al. 2013] future maintenance and evolution of web services.

Anti-patterns detection is therefore important to assess the design of web services and ease their maintenance and evolution. However, methods and techniques for detection authorization constraint anti-patterns in web services orchestrated by BPEL4People do not yet exist. Anti-patterns have many definitions: they are ”poor” solutions to recurring design problems, known solutions for solving problems, which are not practical and usable, and wrong practices that are quite commonly used. Anti-patterns have symptoms and consequences and are induced by some root clauses.

The presence of anti-patterns must be identified in unsuccessful system and their absence shown in successful systems. Anti-patterns usually cause costly and complicated architectures, which are costly and difficult to maintain. Having anti-patterns in a system can hinder a project. Developers study anti-patterns to avoid pitfalls. But sometimes can create pitfalls in knowledge transfer if not applied appropriately. The idea that the use of anti-patterns in knowledge transfer may be a dangerous strategy if applied incorrectly than discovery and corrected before.

## 4. Specification of Authorization Constrain Anti-pattern in Web Services Orchestrated with BPEL4People

In this section we introduce the specification of 7 (seven) authorization constrain anti-pattern. These anti-patterns were adapted from the literature [Smith and Williams 2000], [Moha et al. 2012], [Palma et al. 2014b], [Palma et al. 2014a], [Palma et al. 2013] and [Zha 2006].

- Dangling Inputs Group of Users (DlGU): DIGU is an anti-pattern where one group of users remain unused.
- Dangling Outputs Group of Users (DOGU): DOGU is an anti-pattern where one group of users is assigned for one service but it was not defined.
- Duplicated Group of Users (DGU): DGU corresponds to a group of highly similar users.
- Ambiguous Name of groups of users (ANGU): Ambiguous Name is an anti-pattern where the developers use the names of groups that are very short or long, include too general terms, or even show the improper use of verbs, etc. Ambiguous names are not semantically and syntactically sound and impact the discover ability and the re-usability of a Web service.
- Missing User (MU): When a user that performs a task is excluded to perform another task that he/she is not one user potential of this another task.
- Strongly Missing Groups of Users (SMGU): When who performs a task was the members of one group and another task excludes the members of this group, however no member of this group is a potential user of this another task.
- Weakly Missing Groups of Users (WMGU): When who performs a task was the members of one group and another task excludes the members of this group though some member of this group is a potential user of this another task and others not.

## 5. Approach

With the aim of detecting anti-patterns in authorization constrain in web services orchestrated with BPEL4People, we used the approach shown in 1. This approach involves three major steps:

- Step 1 (Specification) concerns specifying rules for the detection of authorization constrain anti-patterns that, later on, will be applied on web services orchestrated with BPEL4People.
- Step 2 (Generation) transforms orchestrated with BPEL4People into an intermediary representation, i.e., more abstract and simplified, by filtering some process facts those are not required to apply rules to ease: (i) the implementation of the rules defined in the previous step and (ii) the further analysis of the processes.
- Step 3 (Detection) consists in applying the rules defined in Step 1 on the transformed processes in the previous step. Finally, a list of existing anti-patterns with the involved process fragments will be shown.

Static service antipatterns require only static analysis for their detection and, thus, only their structural properties are needed to detect them within the services. Coupled with the information about how these parts will be distributed across the web services orchestrated with BPEL4People, we use rules to detect the potential anti-patterns that may occur. Our approach for detecting anti-patterns leverages static analysis to detecting anti-patterns in authorization constrain in web services orchestrated with BPEL4People.

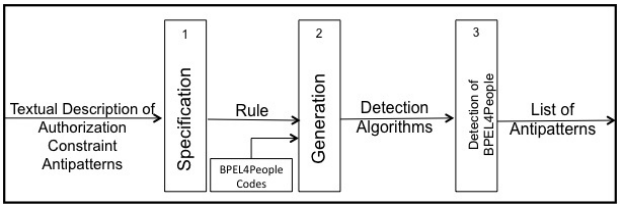



**Figure 1. Proposed Detection Approach.**

### 5.1. Experiments and Results

This section applies the proposed detection approach for seven authorization constraint anti-patterns in Web services orchestrated by BPEL4People anti-patterns, specified in Sections 4.

- Step 1 concerns specifying rules for the detection of authorization constrain anti-patterns that, later on, will be applied on Web services orchestrated with BPEL4People.
- Step 2 transforms orchestrated BPEL4People code into an intermediary representation, i.e., an more abstract and simplified code, by filtering some processes, those are not required to apply rules. This step consists also in implementing the rules showed in 2. We implement the rules in a modularized way, i.e., we implement each side of different logical operators (e.g., AND, OR) in a rule and join them afterwards to check the conformance with the defined conditions using the approach in [8].
- Finally, Step 3 applies the rules defined in Step 1 on the transformed processes in the previous step. The detection step follows the specification of the rules and the

transformation of BPEL4People codes into an intermediary representation, i.e., more abstract and simplified files XML. We show with a small scale experiment the effectiveness of our proposed approach.

We specify seven anti-patterns and we analyze and perform anti-pattern detection of these seven anti-pattern for three Web services orchestrated by BPEL4People using our approach.

- Employee-of-month: this Web services is started and as a first step, the people are determined that qualify as voters for the Employee of the month. Next, all the voters identified before get a chance to cast their votes. After that, the election result is determined by counting the votes casted. After the result is clear, two different people from the set of people entitled to approve the election either accept or reject the voting result. In case any of the two rejects, then there is no Employee of the month elected in the given month, and the process ends. In case all approvals are obtained successfully, the employees are notified about the outcome of the election, and a to-do is created for the elected Employee of the month to prepare an inaugural speech. Once this is completed, the process completes successfully;
- WS-PurchSys: this is a BPEL4People source code for a purchasing process, it is showed in Algorithm 1. Four tasks are defined: manager approve, finance approve, notify staff, and purchase. The potential owners of each task are: manager approve (Alan); finance approve (Ben); purchase (Ben, Cindy, Diana); notify staff (Diana, Edward). The excluded owner of purchase is the actual owner of finance approve. The excluded owner of notify staff is the actual owner of purchase; and
- Web-Service1: this Web services consists of two groups of users (”Group 1” and ”Group 2”) and five services (”Service 1”, ”Service 2”, ”Service 3” (”Service 3” excludes who performed the ”Service 2”), ”Service 4” (”Service 4” excludes who performed the ”Service 1”) and ”Service 5”).

We could detect the seven authorization constrain antipatterns specified in Section 4 in the three Web services orchestrated with BPEL4People given as an examples.

We present in Table 1 the anti-patterns detection results for the three Web services orchestrated with BPEL4People followed by some discussion about accuracy and performance detection. Initially some anti-patterns were not detected in these Web services orchestrated with BPEL4People. Subsequently some changes were inserted in these codes to test the accuracy and performance of the detection algorithm then anti-patterns were detected.

Through our experiment, we aim to show the accuracy and performance of the detection algorithms in terms of precision, recall and detection time, presented in Table 1. Antipattern detection algorithms have at least a precision of 99%. Assuming that the anti-patterns have a negative impact on the design, we target a recall of 100% for anti-patterns, which ensures that we do not miss any existing anti-patterns. The precision concerns the detection accuracy of our specified rules and the corresponding detection algorithms.

The primary threat to the validity concerns the external validity of our results, i.e., generalizing the proposed approach to other Web services orchestrated with BPEL4People. We perform specification and detection of seven authorization constrain anti-patterns. However, we ran the experiment on a set of three Web services orchestrated

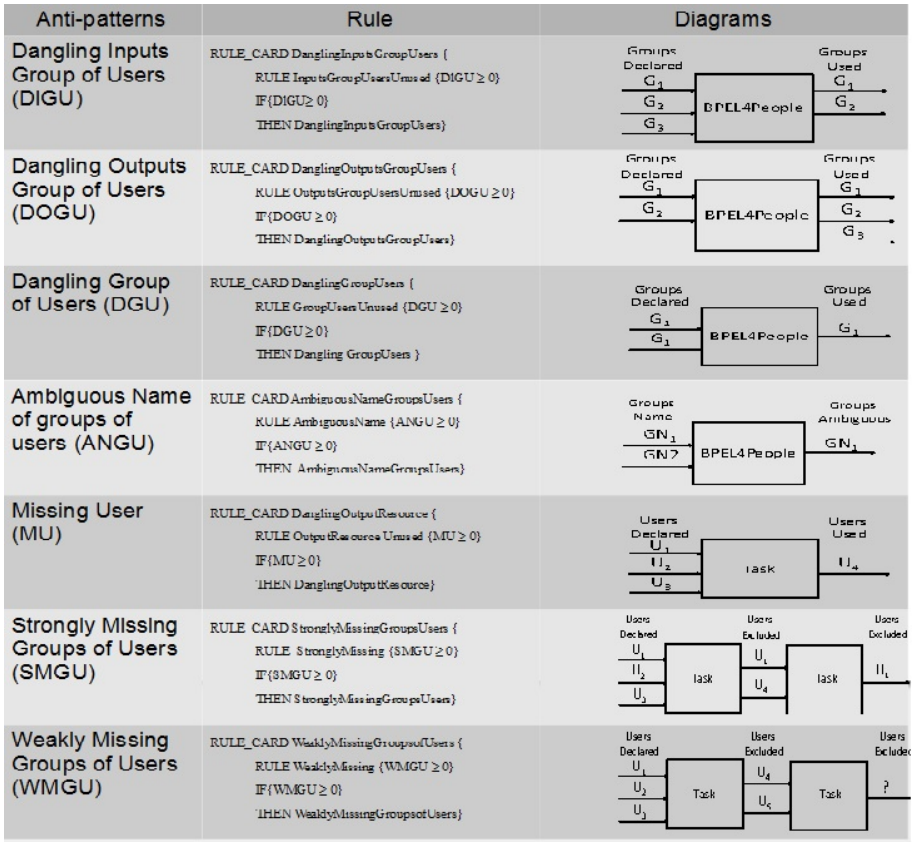



**Figure 2. Rules specifications and diagrams.**

with BPEL4People to minimize this threat. We plan to replicate the experiment on others Web services orchestrated with BPEL4People as a future work. We do not execute the Web services orchestrated with BPEL4People, hence analyze the processes statically, and the injections were performed internally, which are threats to the internal validity. However, we plan to execute the web services orchestrated with BPEL4People flows to dynamically analyze them in the future.

The subjective nature to define rule cards is a threat to construct validity. However, we lessen this threat by defining the rule cards after a thorough literature review. Finally, the conclusion validity threats refer to the relation between the treatment and the out come. We paid full attention not to violate the assumptions of the performed statistical tests. We mainly used non-parametric tests that do not require to make any presumption about the data distribution. The threats to internal validity concern the possibility of replicating this study. To minimize this threat, we provide all the details required to replicate the study, including the source code repositories and the raw data used to compute the statistics on our web services. One major challenge to minimize the threat to the external validity is the very limited availability of open-source Web services orchestrated with BPEL4People.

Finally, a list of anti-patterns, among the seven specified, existing in three Web services orchestrated by BPEL4People is showed in Table 1.

## 6. Conclusions and Future Work

In this paper we assessed the authorization constrain in Web services orchestrated with BPEL4People and eased the maintenance and evolution of these architectures.

**Table 1. Experiments results.**

| ANTIPATTERN | METRIC | PRECISION | RECALL | DETECT. TIME (s) |
|---|---|---|---|---|
| Dangling Inputs Group of Users (DIGU) | NDIGU = 0 | 97% | 100% | 3,34 |
| Dangling Outputs Group of Users (DOGU) | NDOGU = 0 | 98% | 100% | 3,04 |
| Dangling Group of Users (DGU) | NDGU = 0 | 100% | 100% | 3,53 |
| Ambiguous Name of groups of users (ANGU) | NANGU = 0 | 98% | 100% | 3,64 |
| Missing User (MU) | NOU = 0 | 98% | 100% | 3,48 |
| Strongly Missing Groups of Users (SMGU) | NSMGU = 0 | 98% | 100% | 3,21 |
| Weakly Missing Groups of Users (WMGU) | NWMGU = 0 | 100% | 100% | 3,07 |

We proposed an approach to specify authorization constrain anti-patterns and detect them in Web services orchestrated with BPEL4People. We present our static analysis-based approach to detect antipatterns in BPEL4People codes. We have also proposed to assess the design and QoS of Web services orchestrated with BPEL4People. To achieve this goal, we propose an approach to specify authorization constrain anti-patterns and detect them in Web services orchestrated with BPEL4People. Finally, we want to quantify the impact of detection anti-patterns on the maintenance and evolution of Web services orchestrated with BPEL4People.

The contribution of this paper is specifying and detect authorization constrain antipatterns in Web services orchestrated with BPEL4People, by leveraging static code analysis and information about prospective deployment of the application components using Web services orchestrated with BPEL4People. We applied and validated the detection algorithms using three Web services orchestrated by BPEL4People showed in Section 5.1, in terms of precision and recall. We specified and detected seven authorizations constrain antipatterns in an initial experiment with tree Web services orchestrated by BPEL4People. Results showed that this approach has an average detection precision of more than 97% and recall of 100%. Hence, we make a strong recommendation that designers be exposed to antipatterns repeatedly until the pattern is well established. Only then should anti-patterns be used to strengthen the pattern in the designer's mind.

## 7. Acknowledgements

We would like to thank the CNPQ, CAPES, UERN, UFERSA, IFRN, SEDUC-CE and Canada Chair.

## References

(2006). *Verifying BPEL Workflows Under Authorisation Constraints*, volume 4102 of *Lecture Notes in Computer Science*. Springer.

Bertino, E., Crampton, J., and Paci, F. (2006). Access control and authorization constraints for ws-bpel. In *Web Services, 2006. ICWS '06. International Conference on*, pages 275–284.

Bianculli, D., Ghezzi, C., and Spoletini, P. (2007). A model checking approach to verify bpel4ws workflows. In *Service-Oriented Computing and Applications, 2007. SOCA '07. IEEE International Conference on*, pages 13–20.

Dumas, M., van der Aalst, W. M., and ter Hofstede, A. H. (2005). *Process-aware Information Systems: Bridging People and Software Through Process Technology*. John Wiley & Sons, Inc., New York, NY, USA.

H.J.A Holanda, J. Merseguer, G. C. and Serra, A. B. (2010). Performance evaluation of web services orchestrated with ws-bpel4people. In *International Journal of Computer Networks & Communications*, volume 2, pages 117–134. AIRCC Publishing Corporation.

Holmes, T., Vasko, M., and Dustdar, S. (2008). Viebop: Extending bpel engines with bpel4people. In *PDP*, pages 547–555. IEEE Computer Society.

Ings, D., Clément, L., König, D., Mehta, V., Mueller, R., Rangaswamy, R., Rowley, M., and Trickovic, I. (2012). Web services human task (ws-humantask) specification version 1.1. OASIS Committee Specification Draft 12 / Public Review Draft 05.

Moha, N., Palma, F., Nayrolles, M., Conseil, B., Guéhéneuc, Y.-G., Baudry, B., and Jézéquel, J.-M. (2012). Specification and Detection of SOA Antipatterns. In Liu, C., Ludwig, H., Toumani, F., and Yu, Q., editors, *Service-Oriented Computing*, volume 7636 of *Lecture Notes in Computer Science*, pages 1–16. Springer Berlin Heidelberg.

Palma, F., Dubois, J., Moha, N., and Guéhéneuc, Y. (2014a). Detection of REST patterns and antipatterns: A heuristics-based approach. In *Service-Oriented Computing - 12th International Conference, ICSOC 2014, Paris, France, November 3-6, 2014. Proceedings*, volume 8831 of *Lecture Notes in Computer Science*, pages 230–244. Springer.

Palma, F., Moha, N., and Gueheneuc, Y.-G. (2013). Detection of process antipatterns: A bpel perspective. In *Enterprise Distributed Object Computing Conference Workshops (EDOCW), 2013 17th IEEE International*, pages 173–177.

Palma, F., Moha, N., Tremblay, G., and Guéhéneuc, Y. (2014b). Specification and detection of SOA antipatterns in web services. In *Software Architecture - 8th European Conference, ECSA 2014, Vienna, Austria, August 25-29, 2014. Proceedings*, pages 58–73.

Sinnig, D., Gaffar, A., Reichart, D., Forbrig, P., and Seffah, A. (2005). Patterns in model-based engineering. In Jacob, R., Limbourg, Q., and Vanderdonckt, J., editors, *Computer-Aided Design of User Interfaces IV*, pages 197–210. Springer Netherlands.

Smith, C. U. and Williams, L. G. (2000). Software performance antipatterns. In *Proceedings of the 2Nd International Workshop on Software and Performance*, WOSP '00, pages 127–136, New York, NY, USA. ACM.

Smith, C. U. and Williams, L. G. (2003). More new software antipatterns: Even more ways to shoot yourself in the foot. In *29th International Computer Measurement Group Conference, December 7-12, 2003, Dallas, Texas, USA, Proceedings*, pages 717–725. Computer Measurement Group.

# De que Forma a Cultura do Compartilhamento e Modificação pode Colaborar no Processo de Desenvolvimento de Jogos?

**Arison Heltami Rodrigues Uchoa**[1], **Emanuel F. Coutinho**[1]

[1]UFC Virtual – Universidade Federal do Ceará (UFC) – CE – Brasil

arisonheltami@gmail.com, emanuel@virtual.ufc.br

***Resumo.*** *O processo de desenvolvimento de um jogo digital passa por diversas etapas, das mais variadas formas, com o objetivo da criação de um produto final mais eficiente. Em geral, essas etapas são: Conceito, Pré-Produção, Produção e Pós-Produção. Crowdsourcing é um paradigma que está gradualmente sendo usado com mais regularidade e com uma ampla gama de propósitos, tais como reunir muitas fontes de conhecimento, cada uma com diferentes níveis de experiência e especialização, para servir a muitos propósitos. O objetivo deste trabalho é avaliar o impacto do uso de estratégias de crowdsourcing no desenvolvimento e manutenção de engajamento em jogos digitais, delimitando como a construção coletiva enriqueceu o produto.*

## 1. Introdução e Motivação

O advento das redes sociais virtuais e a popularização de diversas plataformas de comercialização e contato entre desenvolvedores e usuários têm promovido, entre outros efeitos, um maior engajamento de participantes externos às equipes de desenvolvimento de jogos em seus processos produtivos. Enquanto alguns tipos de intrusões já são comuns ao processo de desenvolvimento, como por exemplo a busca por *feedback* de usuários e a avaliação de conceitos e interações mediante testes (os chamados *playtests*), tornou-se cada vez mais possível o engajamento da comunidade como um todo na criação de conteúdos para jogos, dispersando os esforços entre um maior número de participantes para a criação de um produto final de qualidade ou derivados que promovem novas experiências partindo do material original.

Segundo Jeff Howe, "Crowdsourcing é atrair pessoas externas a empresa e envolvê-las em processos amplamente colaborativos e criativos" [Howe 2008]. A perspectiva empresarial de Howe também pode ser adequada para projetos individuais e atividades sem fins comerciais. Assim, o *crowdsourcing* combina o esforço do público para resolver um problema ou produzir um recurso [Wang et al. 2013].

O *crowdsourcing* é um paradigma que está gradualmente sendo usado com mais regularidade e com uma ampla gama de propósitos [Carter et al. 2012]. Reunir muitas fontes de conhecimento, cada uma com diferentes níveis de experiência e especialização, através do uso inovador da tecnologia, pode servir a muitos propósitos. O *crowdsourcing* pode ser considerado como a melhoria da terceirização (*outsourcing*), já que a terceirização é quando um trabalho é atribuído a um indivíduo definido, enquanto que o *crowdsourcing* atribui um trabalho a um grupo indefinido e grande de pessoas.

Meios de *crowdsourcing* (colaboração coletiva) estão entre as principais formas de engajar a comunidade ao redor de uma ideia, popularizando-a dentro de um grupo, criando

um maior senso de posse e pertencimento que muitas vezes reflete no resultado comercial do produto, seja pela eliminação ou pela redução de diversos obstáculos técnicos e de mão de obra para o cumprimento de requisitos e etapas necessárias na aplicação.

Esta pesquisa tem como objetivo geral realizar uma avaliação do impacto do uso de estratégias de *crowdsourcing* no desenvolvimento e manutenção de engajamento em jogos digitais, delimitando de que forma a construção coletiva enriqueceu o produto a partir de uma perspectiva qualitativa e comercial. Como objetivos específicos tem-se o mapeamento de estratégias de construção coletiva possíveis para o processo de desenvolvimento de jogos, passando das estratégias mais utilizadas e solidificadas na indústria de desenvolvimento até metodologias com menor amplitude de utilização e o levantamento de projetos que lançaram mão de formas de *crowdsourcing*, especificando os objetivos para a utilização desses meios dentro da proposta do produto final.

Tal pesquisa possui implicações pertinentes para a indústria de desenvolvimento de jogos, com a construção de uma análise estruturada entre os meios e *crowdsourcing*, e o processo de desenvolvimento de um jogo. Ela também pode servir de base para a melhor utilização dessas ferramentas em uma perspectiva mercadológica, com uma maior difusão de formas alternativas de desenvolvimento, propiciando a criação de projetos em caráter experimental, gerando uma maior abertura na indústria de jogos, promovendo a troca de conhecimentos entre partes atuantes nesse setor, do desenvolvimento até o consumo.

## 2. Trabalhos Relacionados

[Carter et al. 2012] discutiram as recentes tecnologias emergentes com foco especial no possível uso do conceito de *crowdsourcing*. Foram discutidos métodos de verificação de qualidade para ajudar no caso do cenário de *design* de jogos de computadores. Também uma arquitetura para permitir o *design* de missões em jogos foi apresentada.

[Munezero et al. 2013] exploraram a plataforma de redes sociais Facebook como fonte de dados textuais com anotações de emoção.Contratar especialistas para fornecer dados manualmente rotulados (anotados) para pesquisa em processamento natural de linguagem é demorado, tedioso e caro. Assim, *crowdsourcing* surgiu como um método útil para obter dados anotados. O trabalho desenvolveu o jogo do Facebook EmotionExpert para coletar dados textuais anotados por humanos e detectar emoções no texto. O jogo fornece um meio para alcançar um grande número de jogadores que realizam anotações do conteúdo emocional dos textos uma atividade social. Descobertas indicam que o jogo é útil para alcançar um grande número de pessoas na produção de anotações confiáveis.

[Shaker et al. 2013] trataram da estética dos jogos de plataforma e o que torna um nível de plataforma envolvente, desafiador ou frustrante. Para isso, utilizou-se mineração de dados de um jogo de *crowdsourcing* de um clone do clássico jogo de plataformas, o Super Mario Bros. Os dados consistem em 40 níveis com variados parâmetros de *design*. Coletivamente, esses níveis são reproduzidos 1560 vezes pela Internet, e a experiência percebida é anotada pelos participantes da experiência.Dada a riqueza desses dados de *crowdsourcing*, à medida que todos os detalhes sobre o comportamento dos jogadores no jogo são registrados, o problema é extrair recursos numéricos significativos no nível apropriado de abstração para a construção de modelos computacionais genéricos de experiência do jogador e, portanto, estética do jogo. Houve um avanço nos fatores que contribuem para a estética do jogo de plataforma, e os resultados foram úteis para a

personalização da experiência do jogo através da adaptação automática.

Com o rápido desenvolvimento do *crowdsourcing* em todo o mundo, o *crowdsourcing* para Engenharia de Software começa a atrair mais atenção de desenvolvedores de software, codificadores e pesquisadores [Hu e Wu 2014]. Muitas plataformas *online* bem-sucedidas demonstraram a capacidade de *crowdsourcing* e o potencial para dar suporte a várias atividades de desenvolvimento de software. No TopCoder, para estudar comportamentos competitivos no *crowdsourcing* de software, aplicou-se teoria dos jogos para modelar desafios de algoritmos, onde codificadores criam desafios. O artigo fornece uma nova perspectiva de pesquisa para o *crowdsourcing* de Engenharia de Software.

[Prandi et al. 2016] apresentaram os resultados obtidos em testes de campo usando o jogo Geo-Zombie, que mistura a realidade com zumbis virtuais em um ambiente urbano. Geo-Zombie é projetado para coleta de dados georreferenciados sobre acessibilidade urbana, com o objetivo de envolver um grande número de jogadores na atividade de sinalização durante caminhadas. Para obter munição para reagir a um ataque de zumbis, os jogadores podem sentir / mapear (*crowdsensing*) barreiras e instalações urbanas, transmitindo esses dados a um centro operacional (*crowdsourcing*). Os resultados confirmam a viabilidade e adequação da abordagem e estimulam discussões interessantes.

[Rojas et al. 2017] apresentou uma abordagem baseada em gamificação e *crowdsourcing* para produzir melhores testes de software. O jogo web CODE DEFENDERS permite que equipes de jogadores disputem em um programa, onde os atacantes tentam criar defeitos sutis, e os defensores tentam contrapor escrevendo fortes testes. Experimentos em cenários controlados em *crowdsourcing* revelam que escrever testes como parte do jogo é mais agradável, e que jogar CODE DEFENDERS resulta em conjuntos de testes mais fortes do que aqueles produzidos por ferramentas automatizadas.

## 3. Relação entre Processos de Produção e Técnicas de *Crowdsourcing*

A relação entre as etapas de desenvolvimento e os meios de *crowdsourcing* foram realizadas partindo de formas possíveis de se aplicar o *crowdsourcing* dentro de cada uma das etapas de desenvolvimento, levando em consideração os artefatos que deveriam ser produzidos durante a etapa específica, o grau de publicização do projeto e as práticas já existentes na indústria de incluir o usuário no ciclo de desenvolvimento.

Um processo de desenvolvimento de um jogo digital passa por diversas etapas [IGN 2006]. Estruturados de modo diferente por diversos autores, empresas e equipes de desenvolvimento, o objetivo dos processos se mantém como etapas que devem ser executadas em ordem para que a criação do produto final seja feita do modo mais eficiente. Comumente esse processo envolve as seguintes etapas:

- Conceito: momento no qual os desenvolvedores estão focados no que é a proposta para o produto, quais seus principais pontos em relação aos concorrentes e sua ideia central.
- Pré-Produção: é o momento no qual a proposta do jogo é validada e os primeiros elementos são criados para se alinharem ideias tanto na implementação quanto nos aspectos artísticos e de experienciais do jogo.
- Produção: é quando ocorre a implementação propriamente dita, com um projeto já estabelecido nas etapas anteriores e objetivos bem definidos para serem cumpridos por todos os profissionais envolvidos.

- Pós-Produção: etapa na qual ocorrem os testes de versões públicas do jogo, adequação aos requerimentos de plataformas específicas, ajustes eventuais de problemas e, por fim, o lançamento.

Desde 2006, quando o termo *crowdsourcing* foi cunhado por Jeff Howe, seu uso vem se tornando cada vez mais abrangente para definir meios de compartilhamento de trabalho e esforço coletivo para atingir um objetivo [Howe 2006]. Tal tática é utilizada em meios mais tradicionais como o de comércio e serviços, assim como em aplicativos. Dentro dessas diversas etapas do processo de desenvolvimento, é possível haver a intrusão de participantes externos ao desenvolvimento, oferecendo colaborações tanto na validação de conceitos apresentados pelos desenvolvedores quanto de forma mais prática, desenvolvendo em conjunto com a equipe o produto final.

Dentro da indústria de jogos, cada uma das etapas do processo de desenvolvimento pode assimilar diversas formas de *crowdsourcing*. Para os desenvolvedores, essas táticas podem se tornar interessantes em detrimento de formas mais tradicionais de conclusão de projetos. Do mesmo modo que a maior parte das empresas que trabalham desenvolvendo produtos para o usuário final, a aplicação do *crowdsourcing* pode refletir os mesmos benefícios levantados, como a redução dos custos de desenvolvimento, resolução de problemas aliado ao *feedback* constante dos usuários, além de ser uma forma de criar uma relação mais próxima entre o usuário e o produto.

Há diversos casos de aplicação de meios de *crowdsourcing* em jogos, os quais foram inseridos no conceito, trabalhados durante o desenvolvimento ou surgiram após o lançamento. Esses casos vão do uso da base de usuários para a definição de temas e conceitos abordados no jogo, algo próprio de fases iniciais do desenvolvimento, passando por *feedbacks* iterativos durante o processo de desenvolvimento mirando a qualidade do produto final até o estabelecimento de meios de modificação e criação baseados no jogo original em um momento posterior ao lançamento do jogo.

Dentro da etapa de Conceito, momento no qual são definidas as principais características do projeto, é muito comum a utilização de meios de questionário e pesquisas de opinião para validar a recepção da ideia diante do público. Da mesma forma que em outras áreas, onde ter certeza que o produto em desenvolvimento possui apelo e satisfaz as necessidades de seu cliente ou público, como o Design Gráfico e o Desenvolvimento de Sistemas, na indústria de jogos tal validação pode decidir todo o futuro do projeto.

Um exemplo de aplicação *crowdsourcing* durante a etapa de Pré-Produção de um jogo está na forma de aproximar os usuários de seu processo de desenvolvimento, criando projetos que possam ser facilmente customizáveis conforme o *feedback* de jogadores por meio de canais de comunicação e mídias sociais.

Durante a Produção de um jogo, como o foco está totalmente na implementação e funcionalidades, o que se pode focar para *crowdsourcing* está na criação de conteúdo e no *feedback* dos usuários sobre o que foi criado. Há práticas comuns na indústria, como a disponibilização de elementos conceituais, *teasers* e testes de conceito produzidos durante o desenvolvimento, especialmente em casos que o financiamento do jogo partiu do público, com o uso de canais de *crowdfunding*, por exemplo, o que são casos em que o desenvolvedor e o consumidor possuem uma relação mais próxima de financiador e prestador de serviço.

A Pós-Produção é a última fase do desenvolvimento de um jogo, envolvendo etapas de teste, validação e lançamento de um projeto, e período pós-lançamento, no qual as empresas costumam manter um suporte ativo ao produto recém-lançado em busca de corrigir problemas e adicionar novos conteúdos. Durante a etapa que o jogo ainda está para ser lançado, normalmente os trabalhos são corretivos e a contribuição que a comunidade oferece é no suporte e validação do jogo por meio de testes.

## 4. Conclusão

Esta pesquisa tem como objetivo realizar uma avaliação do impacto do uso de estratégias de *crowdsourcing* no desenvolvimento e manutenção do engajamento em jogos digitais, considerando etapas gerais do desenvolvimento de jogos, que são: Conceito, Pré-Produção, Produção e Pós-Produção. O estágio da pesquisa ainda é inicial, e tem como próxima etapa uma análise de casos reais de jogos que utilizaram *crowdsourcing* de alguma forma em seu desenvolvimento.

## Referências

Carter, S., Smith, M., Bali, S., Sotiriadis, S., Bessis, N., e Hill, R. (2012). The use of crowdsourcing to aid quest design in games. In *2012 Third International Conference on Emerging Intelligent Data and Web Technologies*, pages 302–305.

Howe, J. (2006). The rise of crowdsourcing — wired. https://www.wired.com/2006/06/crowds/. Online; acessado em março-2018.

Howe, J. (2008). *Crowdsourcing: How the Power of the Crowd is Driving the Future of Business*. Great Britain: Business Books.

Hu, Z. e Wu, W. (2014). A game theoretic model of software crowdsourcing. In *2014 IEEE 8th International Symposium on Service Oriented System Engineering*.

IGN (2006). The game production pipeline: Concept to completion. http://www.ign.com/articles/2006/03/16/the-game-production-pipeline-concept-to-completion. Online; acessado em março-2018.

Munezero, M., Kakkonen, T., Sedano, C. I., Sutinen, E., e Montero, C. S. (2013). Emotionexpert: Facebook game for crowdsourcing annotations for emotion detection. In *2013 IEEE International Games Innovation Conference (IGIC)*, pages 179–186.

Prandi, C., Salomoni, P., Roccetti, M., Nisi, V., e Nunes, N. J. (2016). Walking with geo-zombie: A pervasive game to engage people in urban crowdsourcing. In *2016 International Conference on Computing, Networking and Communications (ICNC)*.

Rojas, J. M., White, T. D., Clegg, B. S., e Fraser, G. (2017). Code defenders: Crowdsourcing effective tests and subtle mutants with a mutation testing game. In *Proceedings of the 39th International Conference on Software Engineering*, ICSE '17, pages 677–688, Piscataway, NJ, USA. IEEE Press.

Shaker, N., Yannakakis, G. N., e Togelius, J. (2013). Crowdsourcing the aesthetics of platform games. *IEEE Transactions on Computational Intelligence and AI in Games*, 5(3):276–290.

Wang, A., Hoang, C. D. V., e Kan, M.-Y. (2013). Perspectives on crowdsourcing annotations for natural language processing. *Language Resources and Evaluation*, 47(1):9–31.

# As competências para atuação na fronteira do conhecimento entre a engenharia de software e as ciências sociais: um ensaio teórico preliminar

**José Jorge Lima Dias Jr.[1], José Adson Oliveira Guedes da Cunha[2]**

[1]Centro de Ciências Sociais Aplicadas – Universidade Federal da Paraíba (UFPB)
João Pessoa – PB – Brasil

[2]Centro de Ciências Aplicadas e Educação – Universidade Federal da Paraíba (UFPB)
Rio Tinto – PB – Brasil.

jorge@dcx.ufpb.br, adson@dcx.ufpb.br

***Abstract.*** *This paper presents an initial theoretical essay aiming to generate reflections on the challenges related to the studies on social and human aspects in Software Engineering (SE). The focus of the discussion is based on the limitations that the SE community has in working on the frontier of knowledge with other areas and how it affects the generation of scientific knowledge. Hence, an epistemological, methodological and theoretical competency model was proposed as a possible solution.*

***Resumo.*** *Este artigo apresenta um ensaio teórico inicial com o objetivo de gerar reflexões acerca dos desafios relacionados aos estudos sobre aspectos sociais e humanos na Engenharia de Software (ES). O foco da discussão baseia-se nas limitações que a comunidade de ES possui em trabalhar na fronteira do conhecimento com outras áreas e como isso afeta a geração do conhecimento científico. Desse modo, um modelo de competências epistemológicas, metodológicas e teóricas é proposto como um possível encaminhamento.*

## 1. Introdução

A comunidade científica em Engenharia de Software (ES) tem reconhecido a importância das questões sociais, cognitivas, culturais, políticas e organizacionais. Como consequência, cada vez mais são aplicadosos conceitos já consolidados da psicologia e das ciências sociais ao contexto de software [Lenberg et al. 2015]. Nesse sentido, a principal tese defendida neste ensaio preliminar é que há uma necessidade *sine qua non* de que os pesquisadores de ES caminhem para a fronteira do conhecimento para compreender os fenômenos oriundos da área e assim gerar avanços significativos e relevantes.

A discussão deste artigo concentra-se nos desafios que um pesquisador da área de ES enfrenta ao se deparar com fenômenos de natureza social e/ou humana, já que normalmente sua formação (graduação e pós-graduação *stricto sensu*) está dentro de uma área considerada de exatas como Ciência da Computação ou equivalente. Desse modo, levantamos o seguinte questionamento: quais competências devem ser desenvolvidas para que esses pesquisadores tenham de fato uma inserção em um campo que toca a fronteira do conhecimento com outras áreas ditas sociais e/ou humanas?

A contribuição deste artigo, portanto, não está vinculada ao rigor metodológico, como acontece normalmente em estudos da área, mas na capacidade de gerar um esforço reflexivo para compreender a realidade. O objetivo é que as reflexões contidas nesse ensaio fomentem a discussão em torno da tese principal, oportunizando a elaboração de ideias e o encaminhamento de possíveis soluções. Essa discussão é especialmente relevante para os programas de pós-graduação *stricto sensu* que possuem linhas de pesquisa interdisciplinar entre a ES e as ciências sociais.

Para atender ao objetivo proposto, o artigo segue a seguinte estrutura: a Seção 2 apresenta as barreiras para se chegar à fronteira do conhecimento; a Seção 3 discute as competências que precisam ser desenvolvidas para romper a barreira do conhecimento; e, por fim, a Seção 4 apresenta as considerações finais e trabalhos futuros.

## 2. As barreiras para se chegar à fronteira do conhecimento

Para analisar a barreira que existe para se acessar a fronteira do conhecimento, nos apoiaremos a uma perspectiva sociológica, especificamente através da lente das comunidades de prática [Wenger 1998]. As comunidades de prática são um conjunto de relacionamentos entre pessoas e suas atividades ao longo do tempo, e são definidas pelos seus membros na forma pelas quais certas coisas são feitas e interpretadas por eles [Bispo 2013]. Dessa forma, podemos perceber o ensino e a pesquisa como uma prática social, cujo os membros são os pesquisadores da área.

A Figura 1 ilustra a ideia central deste artigo. Pesquisadores da área de Ciência da Computação estão cada vez mais buscando compreender fenômenos associados ao seu campo empírico. Muitos desses fenômenos, no entanto, tocam a fronteira com outras ciências já bem estabelecidas. Na figura, cada um dos círculos representa uma comunidade de prática. Desse modo, para acessar o fenômeno, é preciso romper a barreira da fronteira do conhecimento, acessando uma área de interseção com outra área da ciência, ou seja, uma outra comunidade de prática.

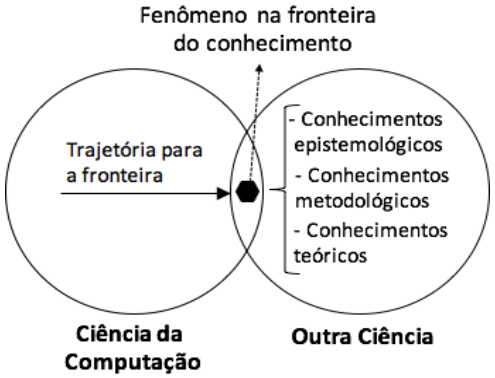



**Figura 1. Trajetória de desenvolvimento de competências para atuar na fronteira do conhecimento**

O conceito de comunidade de prática está diretamente relacionado com a ideia de participação periférica legitimada [Lave e Wenger 1991], que representa a inserção gradual de um novo membro na comunidade, o qual precisa se apropriar do currículo situado pertencente à comunidade. Neste sentido, a legitimação de um pesquisador de fora da comunidade requer uma adaptação, pois ele precisa entender as práticas, a linguagem, a forma de pensamento e de ação de quem faz parte dela.

Desse modo, há a necessidade emergente de se apropriar de algumas competências oriundas dessa ciência outra, permitindo que a geração do conhecimento seja de fato substantiva e que não haja uma duplicação ou repetição de descobertas sobre o fenômeno já bem estabelecidas. Conforme sugerido por Lenberg et al. (2015), é necessário considerar os avanços dessas outras ciências para que não fiquemos "reinventando a roda". Araujo et al. (2015) destacam que esta mudança de perspectiva não é trivial para uma comunidade formada por egressos de pesquisa em uma área fim, como a Computação, e que há a necessidade de se trabalhar fortemente não só na comunidade atual, como também no âmbito da formação de novos mestres, doutores e pesquisadores que irão compor a área futuramente.

Esse artigo, escrito na forma de um ensaio teórico, já pode ser considerado como um esforço em direção à fronteira de interseção com as ciências sociais, pois trata-se de um gênero textual comum nessa área, mas que pode gerar inquietações em áreas como a Computação. No entanto, acreditamos na mutabilidade das comunidades de prática, especialmente a ES, que vem buscando cada vez mais ampliar sua perspectiva em compreender a realidade e em fazer ciência.

## 3. As competências para romper as barreiras da fronteira do conhecimento

Conforme apresentado na Figura 1, para se chegar à fronteira do conhecimento é necessário romper uma barreira de uma outra comunidade científica, e se apropriar de um conjunto de conhecimento oriundo dela. A sugestão trazida aqui é o desenvolvimento de algumas competências, organizadas em três eixos: competências epistemológicas, metodológicas e teóricas.

### 3.1. Competências epistemológicas

As abordagens metodológicas, tanto qualitativas quanto quantitativas, fundamentam-se em pressupostos filosóficos que representam "como" o pesquisador irá aprender e "o que" ele irá aprender com a pesquisa. A dimensão epistemológica relaciona-se ao conhecimento e como ele pode ser obtido [Hirschheim 1992]. Os pressupostos filosóficos são classificados de diferentes formas por vários pesquisadores [Burrel e Morgan 1979; Gephart 2004]. No âmbito da computação, no entanto, há um domínio hegemônico do paradigma funcionalista e uma epistemologia positivista.

A perspectiva epistemológica positivista é fortemente caracterizada pela visão determinista, racional e cartesiana sobre os fatos da realidade, com destaque para a separação excludente entre pesquisador e o objeto de pesquisa [Myers 2005]. O interpretativismo, por sua vez, tem o objetivo de entender o mundo do ponto de vista daqueles que o vivenciam. Nessa abordagem, o objeto de pesquisa é entendido como construído socialmente pelos atores [Schwandt 1994].

Apesar dos esforços para definição de padrões, processos e métodos para o desenvolvimento de software, há de se reconhecer que não estamos lidando com uma ciência exata com leis ou regras pré-estabelecidas, mas com um conjunto complexo de tarefas baseadas nas relações humanas e no conhecimento, experiência, caráter e formação cultural de cada indivíduo. Nesse sentido, apesar de a maior parte das pesquisas na área de computação, e, especificamente, na ES, se concentrar na perspectiva positivista, é importante que os pesquisadores abram seus leques para

investir em métodos que considerem a complexidade dos fatores humanos e das relações sociais no desenvolvimento de software.

### 3.2. Competências metodológicas

A necessidade de se focar no paradigma funcionalista exige a aplicação de métodos de pesquisa usados de forma mais abrangente em outras áreas. A *GroundedTheory* (GT) provou ser uma abordagem de pesquisa útil em vários campos de pesquisa. Por ser um método baseado em um paradigma indutivo diferente do modelo tradicional de pesquisa hipotético-dedutiva na área de computação, seu uso nem sempre é feito de forma adequada. De acordo com uma revisão sistemática da literatura conduzida por Stol et al. (2016), de 98 artigos que mencionam o uso de GT na engenharia de software, apenas 16 fornecem relatos detalhados de seus procedimentos de pesquisa.

O Estudo de Caso, apesar de utilizado há mais tempo na área de computação, nem sempre o é de acordo com a definição clássica de um método empírico com o objetivo de investigar um fenômeno contemporâneo em seu contexto [Runeson 2009]. Pesquisas na área de melhoria de processos e transferência de tecnologia indicam o uso de Estudo de Caso quando deveriam ser Pesquisa-Ação. O primeiro é puramente observacional enquanto o segundo é focado na mudança do processo.

Diante deste cenário, faz-se necessário que os estudantes tenham acesso a pesquisas em outras áreas relacionadas com o fenômeno para entender como os métodos de pesquisa foram utilizados, dada a maior maturidade no uso de tais métodos nas ciências sociais.

### 3.3. Competências teóricas

A utilidade de uma teoria se dá na organização de um fenômeno tendo em vista a previsão de novos fatos e relacionamentos baseados em outros previamente conhecidos, além de indicar pontos que não foram, de forma convincente, explicados [Marconi e Lakatos 2010].

Em programas de Pós-Graduação em computação é esperado que o foco dos conteúdos das disciplinas seja voltado apenas aos aspectos específicos da área. No entanto, algumas disciplinas específicas a uma linha de pesquisa voltada aos aspectos sociais na ES poderiam ser ofertadas. Uma outra alternativa seria permitir que o aluno curse disciplinas em programas de outras áreas do conhecimento.

Ao se pesquisar fenômenos na fronteira do conhecimento, os pesquisadores precisam consultar trabalhos científicos de outra área. Muitas vezes, o completo entendimento das teorias estudadas ou geradas por esses trabalhos dependem do conhecimento sobre epistemologia e metodologia do pesquisador. Se há uma intenção, por exemplo, em se pesquisar aprendizagem organizacional, é necessário absorver conhecimentos de teorias oriundas da Educação, Administração e/ou Psicologia.

## 4. Considerações finais e Trabalhos futuros

Um dos principais desafios no campo da educação em ES é a diversidade de perfis e enfoques dos programas de pós-graduação na área de Ciência da Computação. Isso sinaliza a dificuldade de criação de modelos de formação voltados especificamente a interseção entre ES e as Ciências Sociais. Esse obstáculo acaba gerando dificuldades no

desenvolvimento de competências para pesquisadores que desejam atuar na fronteira do conhecimento da computação com outras áreas, como a Psicologia e Administração. Isso se torna um ciclo vicioso, pois os novos professores e pesquisadores serão aqueles que formarão os futuros alunos.

Como trabalho futuro, os autores vislumbram duas ações: (1) analisar as estruturas curriculares de programas de Pós-Graduação na área de Ciência da Computação para identificar *gaps* nas três dimensões propostas do modelo de competências; e (2) mapear as interseções entre periódicos da Ciência da Computação e periódicos das Ciências Sociais como Psicologia e Administração.

## Referências

Stol, Klaas-Jan; Ralph, Paul; Fitzgerald, Brian. (2016) "Grounded theory in software engineering research: a critical review and guidelines". In: Software Engineering (ICSE), 2016 IEEE/ACM 38th International Conference on. IEEE, 2016. p. 120-131.

Lenberg, P.; Feldt, R.; Wallgren, L. G.. (2015) "Behavioral software engineering: A definition and systematic literature review". Journalof Systems and software, v. 107, p. 15-37.

Wenger, E. (1988), Communities of Practice: learning, meaning and identity. New York: Cambridge.

Lave, J.; Wenger, E. (1991), Situated learning: legitimate peripheral participation. Cambridge: Cambridge University Press.

Bispo, M.(2013) "Aprendizagem organizacional baseada no conceito de prática: contribuições de silviagherardi". Revista de Administração Mackenzie, v.14, n.6, p.132.

Araújo, R. et al. (2005) "A Comunidade de Pesquisa em Sistemas de Informação no Brasil na perspectiva do Simpósio Brasileiro de Sistemas de Informação". iSys-Revista Brasileira de Sistemas de Informação, v. 8, n. 1, p. 5-17.

Hirschheim, R. (1992) "Information Systems Epistemology: An Historical Perspective," in Information Systems Research: Issues, Methods and Practical Guidelines, R. Galliers (ed.), Blackwell Scientific Publications, Oxford, pp. 28-60.

Burrel, G.; Morgan, G. (1979), Sociological paradigms and organizational analysis. London: Heinemann Educational Books.

Gephart Jr., R. P. (2004) "Qualitative Research and the Academy of Management Journal".From the Editors. Academy of Management Journal, Vol.47, No.4,454-462.

Myers, M. D. (1997) "Qualitative Research in Information Systems," MIS Quarterly (21:2), June 1997, pp. 241-242. MISQ Discovery.

Schwandt, T. A. (1994), "Constructivist, Interpretivist Approaches to Human Inquiry". In: Denzin, Norman K. Handbook of qualitative research. Thousand Oaks: Sage.

Runeson, P.; Host, M. (2009) "Guidelines for conducting and reporting case study research in software engineering". Empirical software engineering, v.14, n. 2, p. 131.

Marconi, M. A.; Lakatos, E. M. (2010), Fundamentos de Metodologia Científica, 7 Edição, São Paulo: Editora Atlas.

# Uma Proposta para o Desenvolvimento de Aplicações Orientada a Equipes Interdisciplinares

**Emanuel F. Coutinho**[1], **Leonardo O. Moreira**[1], **Gabriel Paillard**[1]

[1]Instituto Universidade Virtual (UFC Virtual) – Fortaleza – CE
Universidade Federal do Ceará (UFC) – CE – Brasil

{emanuel,leoomoreira,gabriel}@virtual.ufc.br

***Resumo.*** *Em geral, cursos mais técnicos tendem a ter uma grade curricular orientada a produção e tecnologias. Entretanto, devido à globalização e tratamento de fatores técnicos e humanos ao mesmo tempo, adequar essa realidade aos cursos técnicos se torna uma tarefa difícil. Atualmente, a interdisciplinaridade está ganhando espaço, juntando diferentes áreas para um propósito maior. O objetivo deste trabalho é apresentar uma proposta de pesquisa para a integração de equipes com diferentes perfis, focando no desenvolvimento de sistemas diversificados e no desenvolvimento de times interdisciplinares.*

## 1. Introdução e Motivação

Em geral, disciplinas de cursos de graduação no contexto técnico tendem a ter uma abordagem curricular voltada para produção e tecnologias. Comumente, disciplinas, como Engenharia de Software, possuem uma visão direcionada e específica para o desenvolvimento de software com foco no desenvolvedor tradicional. Disciplinas que alinham teoria e prática são normalmente difíceis de se conduzir devido à necessidade de se tratar fatores técnicos (linguagens de programação, ferramentas, componentes etc) e humanos (comunicação, gestão, disponibilidade etc) ao mesmo tempo [Coutinho et al. 2016]. Aspectos de áreas diferentes das que tradicionalmente desenvolvem sistemas, como Saúde e Humanas, estão cada vez mais utilizando recursos de software em suas atividades e pesquisas, implicando na necessidade de uma relação e integração entre as diversas áreas.

Atualmente a interdisciplinaridade está ganhando espaço e unindo áreas diferentes para um propósito maior. Interdisciplinaridade é uma teoria sistematizada comum a um grupo de disciplinas conexas e definidas no nível hierárquico imediatamente superior, o que introduz a noção de finalidade [Japiassu 1976]. É um sistema de dois níveis e de objetivos múltiplos, com coordenação procedendo de um nível superior. A interdisciplinaridade se caracteriza pela intensidade das trocas entre os especialistas e pelo grau de integração real das disciplinas, no interior de um projeto específico de pesquisa. Assim, uma discussão preliminar tem que ocorrer diante da definição do propósito do projeto.

O objetivo deste trabalho é apresentar uma proposta de pesquisa para a integração de equipes de perfis variados, mas com foco no desenvolvimento de sistemas tradicionais (e.g. *web* e *mobile*), com elementos e características não convencionais (e.g. computação física e acessibilidade), promovendo o desenvolvimento de times interdisciplinares.

## 2. Proposta de Pesquisa

A ideia da pesquisa é promover a união mais efetiva de perfis de desenvolvedores tradicionais de sistemas de maneira geral (*web*, *mobile* e banco de dados) com desenvolvedores

com perfil de *design* gráfico digital, com foco em interfaces humano-computador, usabilidade e acessibilidade. A princípio essa junção seria a meta inicial, mas ao longo do tempo novos perfis variados iriam sendo incorporados, principalmente perfis com características de audiovisual, animação digital e sistemas embarcados. Assim, aos poucos uma interdisciplinaridade de fato poderia estar sendo promovida, onde cada um trabalharia um aspecto da aplicação a ser desenvolvida com foco no atendimento de um objetivo maior.

## 3. Metodologia da Pesquisa

Tem-se a intenção de experimentar a proposta em um curso de graduação com características interdisciplinares. Para isso, selecionou-se o curso de graduação em Sistemas e Mídias Digitais, da Universidade Federal do Ceará, com as seguintes etapas planejadas:

- Seleção de Disciplinas: Diversas disciplinas no curso são compatíveis para a pesquisa, pois existem perfis variados tantos dos docentes quanto dos discentes. Neste momento, é necessário selecionar um conjunto de disciplinas que contenham os elementos iniciais da pesquisa: desenvolvimento de sistemas e *design* digital.
- Treinamento dos Docentes: Repasse de alguns conceitos de interdisciplinaridade aos docentes, exemplos, e alguns conceitos de áreas diferentes de formação. Assim, todos os docentes teriam um alinhamento mínimo conceitual sobre outras áreas, facilitando a comunicação para melhor gerenciar as atividades.
- Integração de Atividades: A meta é trabalhar aspectos variados e, de certa forma, sair da zona de conforto. Atividades não comuns de cada disciplina (área) devem ser planejadas. Tanto discentes quanto docentes devem interagir entre si para um melhor fluxo da informação, melhorando a condução das atividades (gerenciais e execução).
- Projetos Integradores: Planejar um projeto integrador, com objetivos bem definidos, que contemple características conceituais e técnicas de todas as disciplinas envolvidas/selecionadas. Esse projeto seria gerenciado pelos docentes, que periodicamente alinhariam às necessidades do projeto e produto, com as necessidades dos discentes.
- Execução e Acompanhamento: Basicamente é o acompanhamento das atividades conforme o planejado. Em caso de replanejamento, é interessante o re-alinhamento com todos da equipe, pois para algumas áreas o impacto pode ser maior.

## 4. Conclusão

Esta pesquisa pretende realizar a junção de profissionais de perfis diferentes de formação, como desenvolvedores e *designers* digitais, em um mesmo projeto de desenvolvimento de aplicações, com características interdisciplinares, para atender um objetivo comum. A ideia é promover uma melhor comunicação, mesmo com perfis variados, preocupando-se com o correto fluxo da informação entre as partes envolvidas. O estudo ainda está em um estágio inicial, mas as possibilidades de se trabalhar de maneira interdisciplinar nas disciplinas do curso selecionado se mostraram promissoras. Estudos preliminares estão ocorrendo, e espera-se que resultados mais concretos estejam disponíveis em breve.

## Referências

Coutinho, E. F., Gomes, G. A. M., e Júnior, A. J. M. L. (2016). Applying design thinking in disciplines of systems development. In *8th Euro American Conference on Telematics and Information Systems (EATIS2016)*.

Japiassu, H. (1976). *Interdisciplinaridade e Patologia do Saber*. IMAGO Editora LTDA, 1 edition.

# Uma proposta de jogo digital para ajudar no combate a fome no Brasil

**Frederico Severo Miranda**[1], **Paulo Cezar Stadzisz**[1]

[1]Universidade Tecnológica Federal do Paraná (UTFPR)
Avenida Sete de Setembro, 3165 – Curitiba – PR – Brasil

fdr.miranda@gmail.com,stadzisz@utfpr.edu.br

***Abstract.*** *Hunger is currently a global problem, reaching 805 million people around the world and 3.4 million brazilians. This is a complex problem that will be eradicated with a better distribution of food and actions of governments and society. The FreeRice web game aims to combat hunger by converting virtual foods into real ones. This article proposes to implement the idea of FreeRice in a 3D virtual platform to help reduce hunger in Brazil.*

## 1. Introdução

A Organização da ONU para Alimentação e Agricultura publicou um relatório que evidencia que uma em cada nove pessoas no mundo sofre com o problema da insegurança alimentar, isso significa que existem 805 milhões de pessoas que não conseguem manter a necessidade diária de alimentação [FAO 2014]. No Brasil, a insegurança alimentar atinge 3,4 milhões de brasileiros e trata-se de um problema complexo que será erradicado com aumento da produção e distribuição de alimentos juntamente com ações e programas de governo [ONU 2014]. Esforços para combater a fome emergem de várias formas, como por exemplo, através dos jogos digitais. O jogo *FreeRice*[1] é um jogo *web* de perguntas e respostas que funciona da seguinte forma: para toda resposta correta, o usuário recebe dez grãos de arroz virtuais. Ao final do jogo[2], o valor acumulado é convertido em alimento real e doado pelos patrocinadores (que possuem anúncios sendo exibidos no site) ao Programa de Alimentação Mundial das Nações Unidas [Jane McGonical 2012]. Os anúncios do *FreeRice* podem ser facilmente bloqueados pelos navegadores atuais, prejudicando o interesse dos patrocinadores e não apresenta os elementos essenciais para manter e atrair novos jogadores. De acordo com [Rouse and Ogden 2010], estes elementos são: interação social, fantasia e imersão.

## 2. Solução proposta

Implementar a ideia do *FreeRice* em um ambiente imersivo que oferece aos jogadores interação social e fantasia. Este ambiente já está sendo desenvolvido como outro projeto dos autores deste artigo, trata-se da Curitiba-ViewPort (C-VP) ilustrada na figura 1. Ressalta-se que, por se tratar de um jogo, não existe a possibilidade de bloquear os anúncios dentro do mesmo. A C-VP é uma cidade virtual 3D multijogador que possui o objetivo de agregar diversas aplicações sérias: planejamento urbano, turismo virtual, aprendizado tangencial, reabilitação cognitiva, redes virtuais 3D etc [Miranda and Stadzisz 2016]. Na C-VP, será criado um conjunto de missões do tipo *combate à fome*. Estas missões terão dois propósitos: (1) conceder moedas virtuais ao jogador,

[1]http://www.freerice.com/

[2]Nao foram encontrados outros jogos semelhantes ao *FreeRice*.

para que o mesmo possa utilizar os recursos da cidade virtual (andar de ônibus, visitar o zoológico, desbloquear novos módulos etc) e (2) conceder alimentos virtuais (grãos de arroz e feijão, unidades de verdura e carne). Estas missões sempre serão relacionadas com alguma marca, como por exemplo: *Visite estabelecimento X* ou *Participe do evento X*. Ao final do mês, o valor dos alimentos virtuais acumulados serão convertidos em alimentos reais e então doados a quem sofre com o problema da insegurança alimentar.

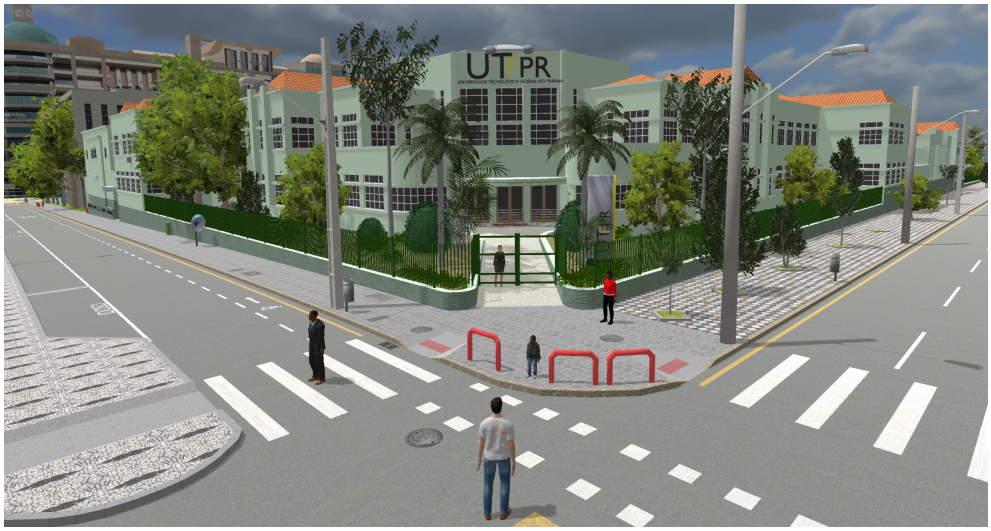



**Figura 1. Curitiba-ViewPort: uma cidade virtual 3D para integrar aplicações. Será adicionado um conjunto de tarefas do tipo *combate à fome* que irá conceder alimentos virtuais ao jogador.Maiores detalhes em https://youtu.be/RPLp4D5N1Ac**

## 3. Considerações Finais

A C-VP está sendo construída para suportar centenas de milhares de pessoas, favorecendo assim dois aspectos: (1) pode se tornar um excelente meio para divulgação de marcas, serviços e interação com clientes, atraindo assim diversos patrocinadores e (2) o pouco esforço de muitos poderá resultar em uma grande quantidade de alimentos. Novas ideias sobre missões do tipo *combate à fome* serão estudadas para motivar e engajar o jogador a realizá-las.Quando a C-VP estiver mais evoluída, tentativas de parcerias com a prefeitura de Curitiba e outros pesquisadores(as) também serão realizadas.

## Referências

FAO (2014). O ESTADO DA SEGURANÇA ALIMENTAR E NUTRICIONAL NO BRASIL Um retrato multidimensional. Technical report.

Jane McGonical (2012). *A realidade em jogo*. 1 edition.

Miranda, F. S. and Stadzisz, P. C. (2016). Curitiba-ViewPort: uma cidade virtual para centralizar aplicações. *SLAT JOGOS - 1º Simpósio Latino-Americano de Jogos*, 2(4):46–51.

ONU (2014). Brasil reduz em 50% o número de pessoas que passam fome. Disponível em: <https://nacoesunidas.org/brasil-reduz-em-50-o-numero-de-pessoas-que-passam-fome-diz-onu>. Acesso em: 10 out. 2016.

Rouse, R. and Ogden, S. (2010). *Game design: Theory and practice*. 2 edition.

# Modelagem de Ecossistemas de Software para Tecnologias Aplicadas a Cursos de Graduação EAD

**Emanuel F. Coutinho**[1], **Italo Santos**[2], **Ernesto Trajano**[1]

[1]UFC Virtual – Universidade Federal do Ceará (UFC) – CE – Brasil
[2]Instituto de Ciências Matemáticas e de Computação
Universidade de São Paulo (ICMC/USP) – São Carlos, SP - Brasil

emanuel@virtual.ufc.br, italo.santos@usp.br, ernesto@virtual.ufc.br

***Resumo.*** *A UAB é um sistema que oferece cursos de nível superior para a população com dificuldade de acesso à formação universitária, por meio da EAD. Diversos cursos de graduação semi-presenciais da UAB utilizam o AVA SOLAR, possuindo cada um diversas disciplinas em cidades distintas, com infraestruturas e tecnologias variadas. O SOLAR ECOS foi modelado para possibilitar uma visão geral de todo o ambiente que circunda o AVA SOLAR. O objetivo deste trabalho é mapear as tecnologias e tipos de recursos multimídia envolvidos nas disciplinas EAD oferecidas no AVA SOLAR.*

## 1. Introdução e Motivação

A Universidade Aberta do Brasil (UAB) é um sistema integrado de universidades públicas que oferece cursos de nível superior para a população com dificuldade de acesso à formação universitária, por meio da Educação à Distância (EAD). A UAB propicia articulação, interação e efetivação de iniciativas que estimulam a parceria dos três níveis governamentais (federal, estadual e municipal) com universidades públicas e demais organizações interessadas, viabilizando mecanismos alternativos para fomento, implantação e execução de cursos de graduação. Atualmente, ela possui no estado do Ceará diversos cursos de graduação, com centenas de alunos atualmente matriculados em diversas cidades, com uma infraestrutura distribuída, espaços físicos cedidos por prefeituras, colégios e instituições federais, e uma plataforma central, o Ambiente Virtual de Aprendizagem (AVA) SOLAR, e uma série de profissionais ligados à EAD.

A Universidade Federal do Ceará (UFC) possui diversos cursos de graduação presenciais e semi-presenciais que utilizam o AVA SOLAR. O SOLAR é um ambiente web cujo modelo de participação é orientado ao professor e ao aluno, possibilitando a publicação de cursos e interação com os mesmos. Também existe uma versão móvel do AVA, porém bem mais simplificada. Seu ECOS foi modelado em [Coutinho et al. 2017].

Cada curso de graduação possui diversas disciplinas em cidades ou polos distintos. Tais polos são diferentes, possuindo infraestruturas variadas. Nem todas as tecnologias de hardware e software necessárias aos cursos estão disponíveis. Além disso, os professores ou tutores podem utilizar diferentes metodologias e aplicações em suas aulas. Nesse sentido, a quantidade de tecnologias utilizada é bem variada, possibilitando novas oportunidades de pesquisa desenvolvimento de aplicações.

Um ECOS pode consistir de um conjunto de atores agindo como uma unidade que interage com um mercado distribuído entre software e serviços, juntamente com as

relações entre estas entidades [Jansen et al. 2009]. Tais relações são frequentemente apoiadas por uma plataforma tecnológica ou por um mercado comum e realizadas pela troca de informação, recursos e artefatos. Nesse contexto, conhecer o ECOS do SOLAR e o que está ao seu redor de tecnologias, não necessariamente vinculados fisicamente a ele, possibilitaria um melhor planejamento e acompanhamento das atividades no AVA.

O objetivo deste trabalho é propor uma pesquisa para o mapeamento das disciplinas EAD oferecidas no AVA SOLAR, suas tecnologias e seus recursos multimídia, e assim modelar um ECOS mais completo e atual, orientado às disciplinas da UAB.

## 2. Proposta de Pesquisa

A pesquisa consiste no mapeamento das disciplinas EAD oferecidas no AVA SOLAR, as ferramentas (software ou hardware) e os recursos multimídia utilizados nas aulas de acordo com os cursos. Para isso, uma série de etapas devem ser executadas:

- Identificar Disciplinas: Muitas disciplinas existem na UAB, ocorrendo ao mesmo tempo e em locais diferentes. Ao se identificar essa lista, pode-se estimar a abrangência que o ECOS SOLAR atinge. Um exemplo é a construção de uma rede sociotécnica.
- Identificar Tecnologias Utilizadas: A mesma disciplina pode utilizar diferentes ferramentas e recursos, seja de hardware ou de software, e de fornecedores diversos. Isto depende muito do local onde a disciplna está ocorrendo, da infraestrutura disponível, das condições de acesso à rede, e dos profissionais envolvidos.
- Identificar Recursos Multimídia: Por recursos multimídia entende-se material produzido e disponibilizado nas aulas das disciplinas que utilizem imagem, som, vídeo e animações. Para a produção desses materiais, são necessários diversos tipos de softwares. Essa etapa consiste em mapear o tipo de material e como foi produzido.
- Modelar ECOS: Gerar modelos de ECOS da UAB relacionado ao softwares utilizados na elaboração das aulas e produção de materiais multimídia. Esse modelo pode ser por polo ou toda UAB. Possivelmente atualizar o ECOS SOLAR [Coutinho et al. 2017] atualmente construído ou delimitar novas fronteiras do ECOS.

## 3. Conclusão

Este trabalho está em um estágio inicial. Espera-se com essa pesquisa que se possa ter uma visão da importância e impacto das tecnologias para as aulas EAD para a comunidade de desenvolvimento e sociedade. Além disso, com uma visão mais global de um ECOS, que seja possível analisar riscos envolvidos no ambiente e melhor planejar as atividades para que elas possam refletir uma maior qualidade no ensino.

## Referências

Coutinho, E. F., Santos, I., e Bezerra, C. I. M. (2017). A software ecosystem for a virtual learning environment: Solar seco. In *2017 IEEE/ACM Joint 5th International Workshop on Software Engineering for Systems-of-Systems and 11th Workshop on Distributed Software Development, Software Ecosystems and Systems-of-Systems (JSOS)*.

Jansen, S., Brinkkemper, S., e Finkelstein, A. (2009). Business network management as a survival strategy: A tale of two software ecosystems. In *Proceedings of the First International Workshop on Software Ecosystems, 11th International Conference on Software Reuse*, pages 34–48.

## Patrocinador Diamante

GOVERNO
DO RIO GRANDE DO NORTE

---

## Patrocinadores Bronze

ESiG
Software & consultoria

dynavideo

CABOTELECOM

Sistema FIERN

PREFEITURA DO NATAL
A NOSSA CIDADE

PARNAMIRIM
Cuidando de você.

---

## Apoio Financeiro

CAPES

MINISTÉRIO DA EDUCAÇÃO

BRASIL
GOVERNO FEDERAL

CNPq
Conselho Nacional de Desenvolvimento Científico e Tecnológico

nic.br
Núcleo de Informação e Coordenação do Ponto BR

cgi.br
Comitê Gestor da Internet no Brasil