

Experiência do Usuário e Engenharia de Software: Interações, Atividades e Produtos

RELATÓRIO TÉCNICO

Ministério da Economia (ME) - Secretaria de Governo Digital (SGD)
Transformação Digital de Serviços Públicos do Governo Brasileiro

Experiência do Usuário e Engenharia de Software:
Interações, Atividades e Produtos

RELATÓRIO TÉCNICO

Elaine Venson

George Marsicano Correa

Andrea Castello Branco Judice

Marcelo Ortega Judice

Wander Cleber Maria Pereira da Silva

Flavio Feitosa Costa

Rejane Maria da Costa Figueiredo

Brasília

2022

Universidade de Brasília

Faculdade UnB Gama

É permitida a reprodução parcial ou total deste relatório, desde que não haja fins comerciais e que seja citada a fonte.

Uma publicação do

Information Technology – Research and Application Center (ITRAC)
Universidade de Brasília
Faculdade do Gama
Área Especial de Indústria – Projeção A
Setor Leste – Gama CEP: 72.444-240

Elaboração

Elaine Venson
George Marsicano Correa
Andrea Castello Branco Judice
Marcelo Ortega Judice
Wander Cleber Maria Pereira da Silva
Flavio Feitosa Costa
Rejane Maria da Costa Figueiredo

Coordenação do projeto

Profa. Dra. Rejane Maria da Costa Figueiredo

E96

**Experiência do usuário e engenharia de software [recurso eletrônico] : interações, atividades e produtos : relatório técnico / Elaine Venson ... [et al.]. – Brasília : Universidade de Brasília, Faculdade UnB Gama, 2022.
32 p. : il.**

Inclui bibliografia.

Modo de acesso: World Wide Web.

1. Experiência do usuário. 2. Engenharia de software. 3. Engenharia de requisitos. I. Venson, Elaine.

CDU 004.5

Lista de abreviaturas e siglas

ER	Engenharia de Requisitos
ES	Engenharia de Software
PO	Product Owner
UCD	User-Centered Design
UX	User Experience
UXD	User Experience Design

Sumário

1	INTRODUÇÃO	5
2	CONCEITOS	6
2.1	Experiência do Usuário (UX)	6
2.2	UX Design (UXD)	7
2.3	Design Centrado no Usuário (UCD)	7
3	IMPACTOS DA UX NA ENGENHARIA DE REQUISITOS	8
3.1	O papel da UX na Engenharia de Requisitos	8
3.2	UX como Requisito	8
3.3	Desafios da adoção de UX no processo de ER	10
3.4	Propostas de incorporação de UX na ER	11
4	INTEGRAÇÃO DO DESIGN DE UX NO DESENVOLVIMENTO DE SOFTWARE	13
4.1	Framework Scrum-UCD	13
4.2	Modelo Funil de UX Ágil	16
4.3	Abordagem Combinada de Desenvolvimento Ágil, UCD e Lean Startup	20
4.4	User-Centered Behavioral (UCB) Software Development Model	22
5	INTEGRAÇÃO ENGENHARIA DE REQUISITOS E EXPERIÊN- CIA DO USUÁRIO NO GPS	25
5.1	O Guia para o Desenvolvimento de Produtos e Soluções	25
5.2	Análise da Integração REUX no contexto do GPS	26
6	CONSIDERAÇÕES FINAIS	29
	REFERÊNCIAS	31

1 Introdução

No escopo do projeto *Transformação Digital de Serviços Públicos do Governo Brasileiro*, oriundo da parceria em Pesquisa e Desenvolvimento firmada entre a Universidade de Brasília (UnB) e o Ministério da Economia (ME), uma das frentes de trabalho atua de forma a apoiar o ministério na formulação de soluções a partir da pesquisa científica e técnicas de Engenharia de Software e de Design da Experiência do Usuário, visando melhorar a qualidade de seus processos, produtos e serviços e contribuir para a inovação. Neste contexto, uma das atividades realizadas teve como objetivo examinar a literatura existente com relação às interações entre as áreas de conhecimento de Experiência do Usuário (UX), de Engenharia de Requisitos (ER) e de Engenharia de Software (ES).

A motivação para este estudo se deu em função de diferentes perspectivas entre pesquisadores, servidores da Secretaria de Governo Digital (SGD) do ME e membros das startups vinculadas ao programa Startup Gov.br, acerca de conceitos, atividades e produtos envolvidos nos processos de UX e ES. De um ponto de vista prático, tem-se percebido um sombreamento entre atividades realizadas pelos designers de UX e pelos engenheiros de software nos projetos desenvolvidos.

Este relatório apresenta o resultado de um estudo bibliográfico que buscou extrair das bases científicas informações acerca do estado da arte a respeito dos processos e interações entre a UX e o desenvolvimento de software. Sendo a UX um conceito de design empregado em diversas áreas, a consulta bibliográfica apresenta-se na perspectiva da ER e da ES e como estas áreas estão sendo impactadas com a adoção de práticas de UX.

O relatório também traz uma análise da integração entre as atividades da ER e da UX à luz do Guia para o Desenvolvimento de Produtos e Soluções (GPS), desenvolvido também no âmbito da parceria entre o ME e a UnB. A partir deste estudo, espera-se oferecer uma base de conhecimentos para a melhoria contínua de processos e produtos em elaboração dentro da SGD.

Este relatório está organizado da seguinte forma: o Capítulo 2 apresenta os principais conceitos relacionados a UX tratados neste relatório; no Capítulo 3 são apresentados os impactos da UX na disciplina de Engenharia de Requisitos; o Capítulo 4 traz propostas de integração da UX no processo de desenvolvimento ágil de software e no Capítulo 5 uma análise do sombreamento entre as atividades de ER e UX no GPS; por fim, o Capítulo 6 traz as considerações finais sobre esta revisão de literatura.

2 Conceitos

Este capítulo aborda as definições a respeito da experiência do usuário e termos relacionados.

2.1 Experiência do Usuário (UX)

Experiência do usuário ou UX, do inglês *user experience* é “a soma das emoções, percepções e reações que uma pessoa experimenta ao interagir com um produto ou serviço” (TOSI, 2020). Para Benyon (2019), a experiência do usuário está relacionada com o desenvolvimento de sistemas interativos, produtos e serviços de alta qualidade que se adaptam às pessoas e ao seu modo de vida.

Diariamente, interagimos com produtos e serviços, passando por experiências diversas. Por exemplo, ao realizar uma compra online, desde o momento em que acessamos o serviço até concluirmos a compra e recebermos o produto, podemos experimentar sensações, pensamentos e emoções que podem ser positivos, negativos ou neutros. O resultado dessa experiência pode fazer que com voltemos a comprar na mesma loja, com que sejamos promotores da marca ou, por outro lado, com que a evitemos e sejamos detratores da marca.

A ISO/IEC 9241-210:2010 define UX como “as percepções e respostas de uma pessoa que resultam do uso ou uso antecipado de um produto, sistema ou serviço” (MIRNIG et al., 2015). Esta definição é complementada na norma com três notas explicativas:

1. A experiência do usuário inclui múltiplos aspectos referentes ao usuário tais como emoções, crenças, preferências, percepções, respostas físicas e psicológicas, comportamentos e conquistas que ocorrem antes, durante e após o uso;
2. A experiência do usuário é uma consequência de muitos fatores ligados ao sistema em si (imagem da marca, apresentação, funcionalidade, performance etc), ao estado interno e físico do usuário (resultante de experiências anteriores, atitudes, habilidades e personalidade) e ao contexto de uso;
3. Quando a usabilidade é interpretada da perspectiva dos objetivos do usuário, ela pode incluir aspectos que permitem avaliar a experiência do usuário.

A importância de se aplicar UX nos mais variados tipos de produtos e serviços é muitas vezes ilustrada utilizando-se exemplos do alto custo de sua *não* utilização. Um

projeto da área de arquitetura que não considera a UX pode, por exemplo, produzir espaços difíceis de se conviver que perduram por muito tempo; no design de carros, pode levar a acidentes graves ou até fatais; em sistemas da área médica pode levar a falhas na administração de medicamentos causando graves problemas (HARTSON; PYLA, 2018).

No contexto do desenvolvimento software, consequências da não aplicação dos princípios de UX podem ser a não utilização do produto ou serviço, a troca pela concorrência, maiores custos de treinamento e suporte aos usuários, ineficiência de uso, insatisfação dos usuários, erros operacionais etc.

2.2 UX Design (UXD)

Se, conforme as definições de UX, a experiência é algo interno ao usuário, pode-se perguntar como é possível fazer o design dessa experiência. Hartson e Pyla (2018) explicam que o termo *design de UX* (UXD) analisado dessa forma parece não fazer sentido, mas pode-se pensar no UXD como o design *para* a experiência do usuário.

Considerando que a experiência do usuário desenvolve-se na interação com o produto ou serviço, o trabalho de um designer de UX é então fazer o design dessa interação de forma que ela seja produtiva, gratificante, satisfatória e até mesmo alegre (HARTSON; PYLA, 2018).

Para isso, a área de Design de UX vem estabelecendo uma série de atividades, métodos, técnicas e produtos que permitem definir de forma estruturada e gradual o design de um produto que esteja melhor adaptado às necessidades do usuário e da marca (TOSI, 2020).

2.3 Design Centrado no Usuário (UCD)

O design de UX baseia-se no conceito de design centrado no usuário ou UCD, do inglês *user-centered design*. UCD é um termo cunhado por Don Norman em 1980, o qual estabelece uma abordagem para criação de produtos que são efetivos, eficientes e usáveis (ANITHA; PRABHU, 2012). O UCD coloca as necessidades e desejos do usuário como elementos centrais do início ao fim do processo de design (STILL; CRANE, 2017).

A abordagem UCD vem evoluindo ao longo do tempo, reconhecendo o papel cada vez mais ativo dos usuários no desenvolvimento de produtos. Assim, os usuários que antes eram vistos apenas como meros repositórios de informação, passaram a ser valorizados por suas experiências e a envolverem-se cada vez mais ativamente em todas as fases do desenvolvimento de um produto (TOSI, 2020).

3 Impactos da UX na Engenharia de Requisitos

A UX muda o foco de atenção dos produtos e funções para os aspectos humanos e sentimentos. Essa característica traz novas perspectivas para o desenvolvimento de software. Neste capítulo são abordados os impactos da UX na disciplina de Engenharia de Requisitos.

3.1 O papel da UX na Engenharia de Requisitos

No contexto do desenvolvimento de software, o Design de UX (UXD) e a Engenharia de Requisitos (ER) possuem alguns propósitos e atividades em comum. Ambas disciplinas elicitam informações de stakeholders e buscam entender como os usuários irão interagir com o sistema (OHASHI et al., 2018).

Em complemento à ER, o design de UX propõe uma amplo conjunto de métodos e técnicas que permite explorar aspectos que influenciam a aceitação do sistema pelos seus usuários, tais como a própria experiência do usuário ao utilizar o software, suas respostas emocionais e afetivas, sentimentos e sensações (KASHFI et al., 2016; SUTCLIFFE, 2016; KHARSHILADZE; LUO, 2015).

Além de buscar compreender o ecossistema do usuário para desenvolver produtos mais efetivos, o designer de UX também tem um papel importante em estabelecer um entendimento a respeito dos segmentos de usuários e da competitividade do mercado na qual o produto estará inserido (ANITHA; PRABHU, 2012).

3.2 UX como Requisito

Em geral, os estudos na área de ER abordam a experiência do usuário como um requisito de qualidade que precisa ser considerado ao longo do ciclo de vida do desenvolvimento de software de forma a aumentar a satisfação do usuário. (KHARSHILADZE; LUO, 2015; KASHFI et al., 2016; OHASHI et al., 2018; ANITHA; PRABHU, 2012).

Kashfi et al. (2016) apresentam um modelo conceitual de requisitos voltados para UX, buscando mostrar as inter-relações entre UX e requisitos funcionais e de qualidade. Para estes autores, a área de requisitos tem um papel importante na prática efetiva de UX, uma vez que se o design de UX não for incorporado aos documentos de requisitos,

acabará sendo negligenciado. Para isso, uma terminologia comum é proposta a partir do mapeamento de conceitos e modelos de UX para modelos e padrões em Engenharia de Software (ES).

O modelo proposto por [Kashfi et al. \(2016\)](#) é apresentado na Figura 1. A pirâmide invertida enfatiza que as camadas superiores emergem e dependem dos requisitos da camada abaixo. Os requisitos de UX, por exemplo, estão posicionados na parte superior da pirâmide, indicando que a percepção do usuário é definida pelos requisitos de UX, mas essa percepção também dependerá dos requisitos de qualidade (QR) e requisitos funcionais (FR) estabelecidos abaixo.

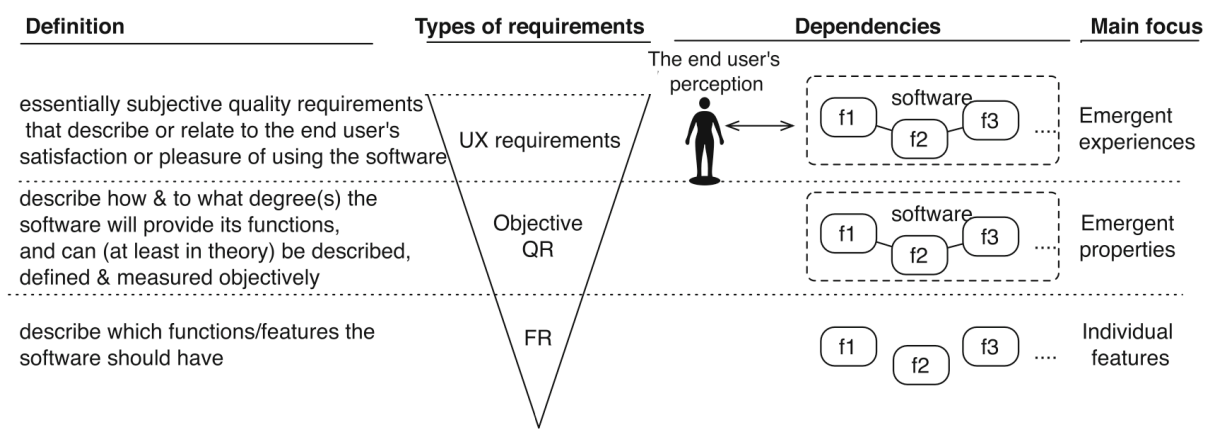


Figura 1 – Fonte: ([KASHFI et al., 2016](#))

O modelo também separa requisitos de qualidade (QR) em objetivos e subjetivos. Requisitos de qualidade objetivos são aqueles que podem ser medidos ou testados objetivamente. Os requisitos de qualidade subjetivos são aqueles que envolvem a percepção subjetiva do usuário e são denominados pelos autores de requisitos de UX ([KASHFI et al., 2016](#)). O modelo, dessa forma, busca situar os requisitos de UX em relação aos requisitos funcionais e outros requisitos de qualidade.

Conforme indicado por [Kashfi et al. \(2016\)](#), na prática, os requisitos de UX precisam ser refinados em requisitos funcionais e requisitos de qualidade objetivos. No entanto, os autores reconhecem que essa tradução dos requisitos de UX em requisitos mais concretos e em soluções de design é ainda difícil e as orientações são limitadas. Aliado a isso, a experiência do usuário com um software pode mudar com o tempo, trazendo ainda mais desafios em sua avaliação.

Para [Hartson e Pyla \(2018\)](#), os designers de UX definem seus próprios requisitos de design. Estes requisitos são em geral traduzidos em histórias de usuário adicionais e protótipos a serem utilizados pelos desenvolvedores de software.

Com base em uma revisão sistemática da literatura, [Kharshiladze e Luo \(2015\)](#) analisaram como o a UX afeta o processo de ER. De acordo com eles, três aspectos de UX

podem ser capturados durante o processo de eliciação de requisitos:

- Usabilidade;
- Fluxo e estética;
- Emoção.

A *usabilidade* é considerada como um requisito de qualidade na ER, mas em geral é tratada tardiamente apenas na fase de arquitetura e projeto do software. Para melhorar a experiência do usuário, [Kharshiladze e Luo \(2015\)](#) sugerem que estudos de usabilidade deveriam ser realizados no início do ciclo desenvolvimento de forma a capturar os requisitos de usabilidade necessários ao produto e direcionar a concepção das funcionalidades.

Fluxo e estética estão relacionados em como a aparência e o design do produto afetam a percepção e o interesse do usuário. A eliciação da experiência do usuário nestes aspectos ajuda a entender como engajar o usuário no uso do produto, oferecendo-lhe uma experiência estética agradável ([KHARSHILADZE; LUO, 2015](#)).

A inclusão de requisitos de *emoção* na ER permite antecipar respostas emocionais do usuário e planejar elementos de design que o façam ter uma atitude positiva e de contentamento em relação ao produto. Neste sentido o requisito deve fornecer informações sobre o estado emocional alvo e o contexto artístico ([KHARSHILADZE; LUO, 2015](#)).

3.3 Desafios da adoção de UX no processo de ER

Várias dificuldades são elencadas na adoção de UX com a ER ou a Engenharia de Software em geral. Um desafio inicial é a falta de conhecimento e compreensão do que é o design de UX ([KASHFI et al., 2016](#); [ANITHA; PRABHU, 2012](#)). Com isso, frequentemente o design de UX ocorre tardiamente no processo de desenvolvimento, com atuação limitada ao design de interfaces ([ANITHA; PRABHU, 2012](#)).

Outro problema frequentemente mencionado na literatura é a falta de especialistas UX para atuar de projetos de software. Engenheiros de requisitos vêm desempenhando algumas das atividades destes profissionais, mas muitas vezes sem o conhecimento e formação adequados e gerando resultados não tão bons ([OHASHI et al., 2018](#)).

Outra questão ainda é a natureza subjetiva da UX - variáveis emocionais, afetivas, hedônicas e estéticas não são facilmente traduzidas em requisitos de software e ainda existem poucos mecanismos disponíveis para isso ([KHARSHILADZE; LUO, 2015](#); [ANITHA; PRABHU, 2012](#)). Além disso, ferramentas que abordam a experiência do usuário, como por exemplo seu estado emocional antes, durante e após o uso do software, estão ainda limitadas ao contexto acadêmico ([KHARSHILADZE; LUO, 2015](#)).

3.4 Propostas de incorporação de UX na ER

Dado o crescente interesse em aplicar UX em projetos de software, estudos na área de Engenharia de Requisitos tem buscado formas de incorporar métodos de UX nas atividades de requisitos.

Sutcliffe (2016) propõe um modelo e um processo para analisar o papel da emoção em aplicações interativas, centradas no usuário, no contexto de interfaces baseadas em personagens/agentes. Ohashi et al. (2018) propõem uma abordagem para coletar e calcular uma pontuação UX de um produto, permitindo identificar problemas de UX como parte das atividades de Engenharia de Requisitos.

Anitha e Prabhu (2012) apresentam um framework de integração entre ER e UX, no qual destacam os pontos em que especialistas em ER e UX devem cooperar. A Figura 2 ilustra as atividades e os pontos de colaboração.

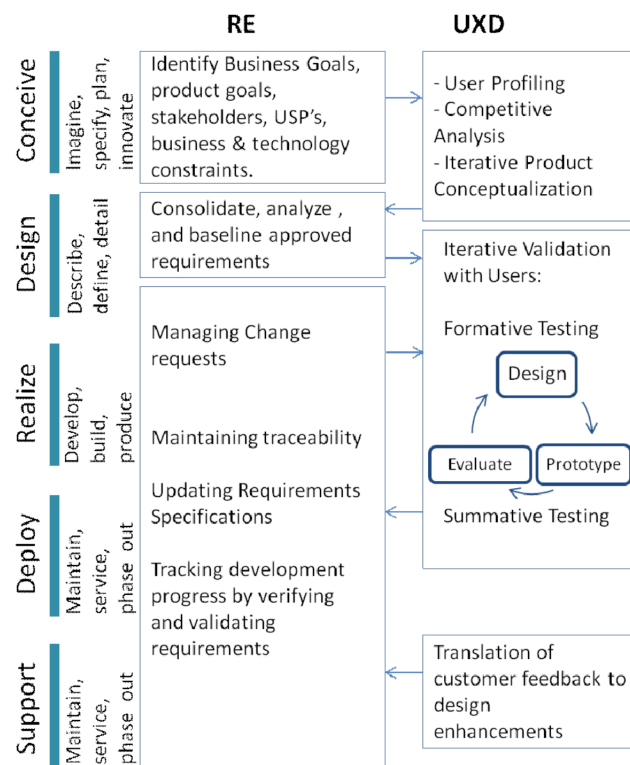


Figura 2 – Integração entre RE e UXD. Fonte: (ANITHA; PRABHU, 2012)

Na fase de *concepção* (*Conceive*), o especialista em ER compartilha com o especialista em UX as informações levantadas a respeito dos objetivos de negócio, posicionamento do produto e stakeholders. Tais informações irão ajudar o especialista em UX em seu estudo sobre os usuários e como estes se beneficiarão com o produto.

Na fase de *design* a viabilidade dos requisitos é analisada em conjunto com o conceito do produto definido pelo especialista UX.

Na fase de *construção (Realize)*, o especialista de requisitos e o especialista UX trabalham alinhados para fornecer explicações e informações detalhadas para o time de desenvolvimento. Ao longo dessa fase e quando o produto já estiver em produção, ambos os especialistas interagem para lidar com as mudanças de requisitos relacionadas a UX e também para validar os requisitos já implementados junto aos usuários.

Na fase de *suporte (Support)*, os especialistas em ER e UX trabalham juntos para coletar respostas e opiniões dos usuários, comunicando-as ao time de desenvolvimento.

4 Integração do design de UX no Desenvolvimento de Software

A indústria de software vem percebendo a necessidade e os benefícios da aplicação do design de UX (UXD). Na literatura acadêmica, o UXD vem sendo abordado em estudos recentes, muitos deles empíricos, que apresentam estudos de caso e propostas de modelos e frameworks de integração entre o desenvolvimento ágil de software e a UX. Este capítulo apresenta algumas dessas propostas.

4.1 Framework Scrum-UCD

[Argumanis, Moquillaza e Paz \(2021\)](#) propõem um framework Scrum-UCD baseado no desenvolvimento do design em paralelo com as sprints de implementação. Embora trabalhando em paralelo, os desenvolvedores participam da validação dos protótipos e os designers participam de todas as cerimônias do Scrum, visando proporcionar uma boa comunicação entre os participantes.

O framework é composto e três fases, Iniciação, Planejamento e Implementação, ao longo das quais são aplicadas nove técnicas de UX e Desenvolvimento Ágil([ARGUMANIS; MOQUILLAZA; PAZ, 2021](#)):

- Protótipos em papel;
- Personas;
- Sprint 0;
- *Pair designing*;
- *Card sorting*;
- Avaliação heurística;
- Pesquisa contextual (*contextual inquiry*);
- *Thinking aloud*;
- Mapeamento da jornada do cliente.

Na sprint 0, fase de *Iniciação*, o Product Owner (PO) define a visão do projeto, que será posteriormente refinada, após uma pesquisa contextual com os usuários reais do

produto. Ainda nesta etapa, os designers, ou especialistas UCD, criam as Personas com base nas informações obtidas na pesquisa conceitual. Caso seja um projeto de redesign, uma avaliação heurística é realizada para identificar os problemas a serem resolvidos e as Personas existentes são atualizadas. Por fim, o backlog do produto é criado, validado pelos usuários e melhorado continuamente (ARGUMANIS; MOQUILLAZA; PAZ, 2021). Este processo está representado nas Figuras 3 e 4, em diagramas BPMN.

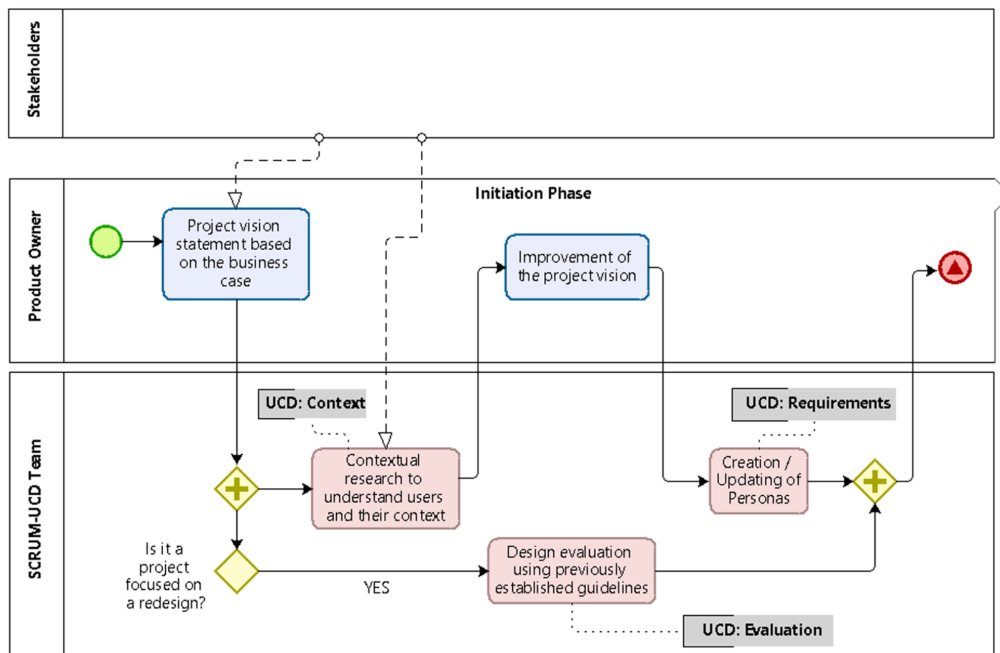


Figura 3 – Fase de iniciação do projeto. Fonte: (ARGUMANIS; MOQUILLAZA; PAZ, 2021)

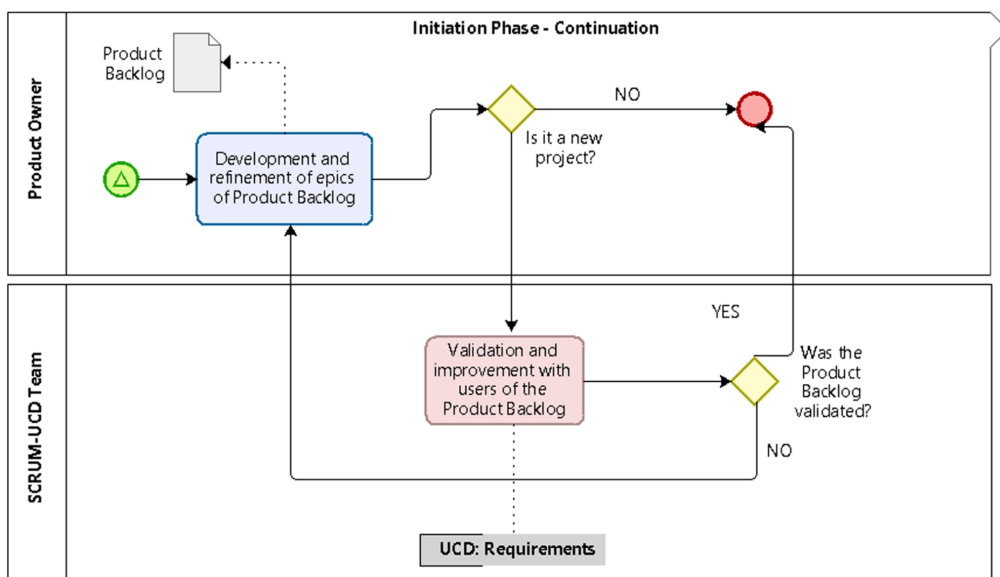


Figura 4 – Fase de iniciação do projeto (continuação). Fonte: (ARGUMANIS; MOQUILLAZA; PAZ, 2021)

Na sequência, ainda na sprint 0, ocorre a fase de *Planejamento*. Esta etapa pode incluir o uso de *card sorting* para criar ou redefinir a estrutura de informação do sistema e o *mapeamento da jornada do cliente* para definir os fluxos que os usuários seguirão no sistema. Com base nessas informações, as histórias de usuário são criadas e estimadas e os designers iniciam a criação dos protótipos, levando em conta as restrições técnicas informadas pelos desenvolvedores. Em paralelo, os desenvolvedores definem a arquitetura do sistema e as ferramentas a serem utilizadas. Ao fim dessa fase, as tarefas são identificadas e incluídas no backlog da sprint. Este backlog serve tanto para a sprint de design quanto para a sprint de desenvolvimento (ARGUMANIS; MOQUILLAZA; PAZ, 2021). As atividades dessa fase estão representado nos diagramas das Figuras 5 e 6.

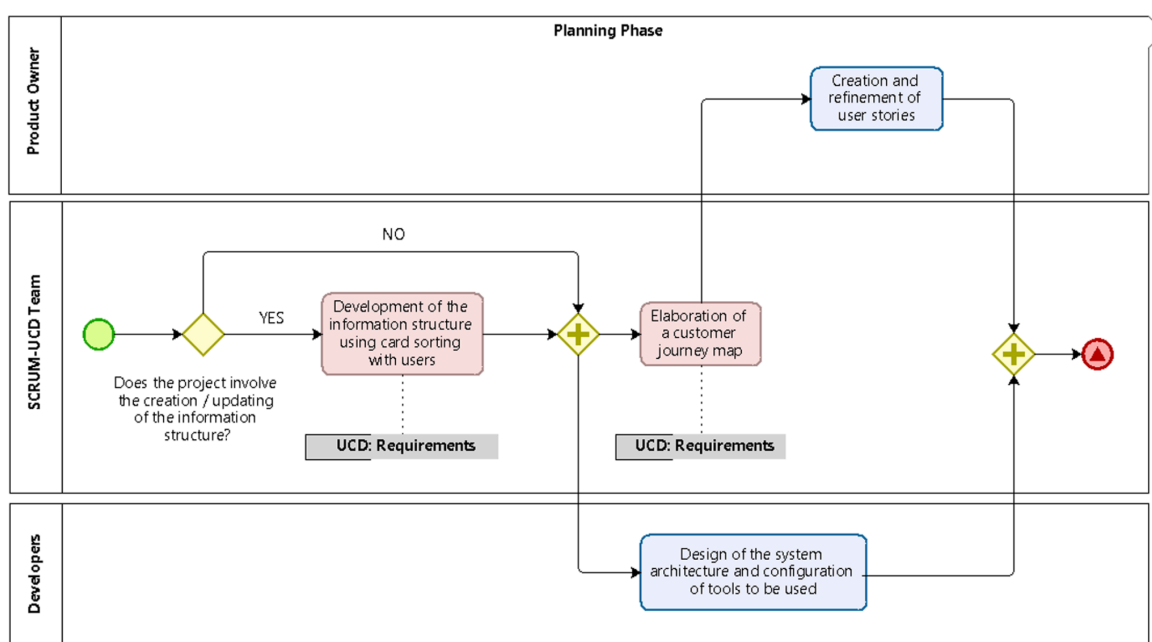


Figura 5 – Fase de planejamento do projeto. Fonte: (ARGUMANIS; MOQUILLAZA; PAZ, 2021)

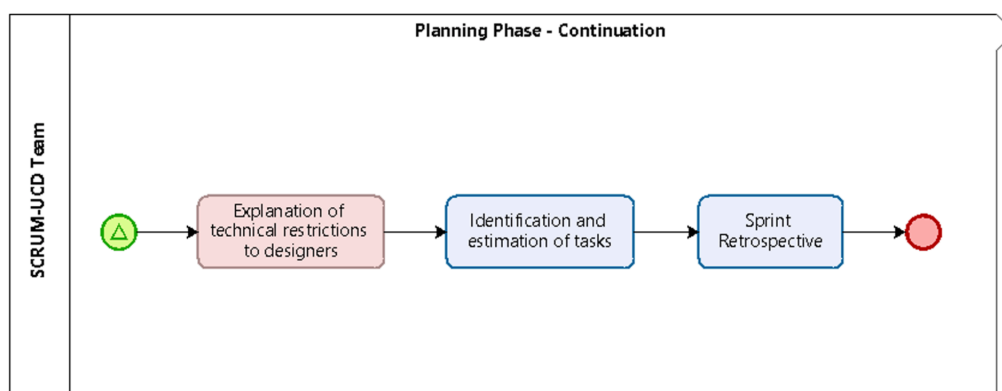


Figura 6 – Fase de planejamento do projeto (continuação). Fonte: (ARGUMANIS; MOQUILLAZA; PAZ, 2021)

A fase de *Implementação* ocorre com as sprints de design e desenvolvimento realizadas em paralelo. De acordo com o framework, as sprints de design estarão sempre uma iteração à frente das sprints de desenvolvimento. A primeira sprint de desenvolvimento é utilizada para estruturar o *back-end* e a base de dados. As sprints de design produzem os protótipos de baixa e alta fidelidade, os avaliam com os desenvolvedores através do processo de *pair designing* e os validam continuamente com o PO e usuários finais. Ao final de cada sprint, os entregáveis de design e o software implementado são validados pelo PO e usuários e os ajustes identificados como necessários são incluídos no backlog das próximas iterações. Por fim, é realizada a reunião de retrospectiva com todos os envolvidos (ARGUMANIS; MOQUILLAZA; PAZ, 2021). As atividades da fase de implementação estão representado nos diagramas das Figuras 7 e 8.

O framework foi testado em um projeto real de desenvolvimento de software com quatro sprints de uma semana. Os resultados reportados compararam o uso do Scrum-UCD com o uso do Scrum típico na mesma organização. De acordo uma avaliação que usou a técnica *thinking aloud* com cinco diferentes usuários, o Scrum-UCD teve um impacto positivo na usabilidade do produto final e melhora nos custos de desenvolvimento, devido à diminuição dos riscos de encontrar problemas de design em interfaces já implementadas. No entanto, eles observaram que sprints de uma semana foram muito curtas para completar todas as atividades requeridas de UCD, sugerindo então o planejamento de sprints com a duração de pelo menos duas semanas (ARGUMANIS; MOQUILLAZA; PAZ, 2021).

4.2 Modelo Funil de UX Ágil

Hartson e Pyla (2018) em seu livro “The UX Book: Agile UX Design for a Quality User Experience” apresentam um processo o qual denominam de *Modelo Funil de UX Ágil* (tradução nossa). Assim como no ciclo de vida de um processo ágil de desenvolvimento de software, o modelo de UX ágil gerencia as mudanças de design que ocorrem durante o processo, entregando designs de UX em pequenos pedaços (HARTSON; PYLA, 2018).

Para Hartson e Pyla (2018) um entrave à integração entre UX e o desenvolvimento ágil é que enquanto o ágil busca criar um modelo responsivo às mudanças ao longo do processo fracionando o desenvolvimento em ciclos de curta duração (as sprints), o design de UX atua de forma mais holística em um escopo maior. O design da UX funciona dessa forma porque as avaliações do usuário ocorrem com alta frequência durante o todo processo de design (HARTSON; PYLA, 2018). Tendo em vista essa divergência, os autores propõe o *modelo funil de UX ágil*, como uma forma de organizar as atividades de design de UX em dois momentos: antes e após sincronizar com a execução das sprints. A Figura 9 apresenta o modelo funil de UX ágil, adaptado de Hartson e Pyla (2018).

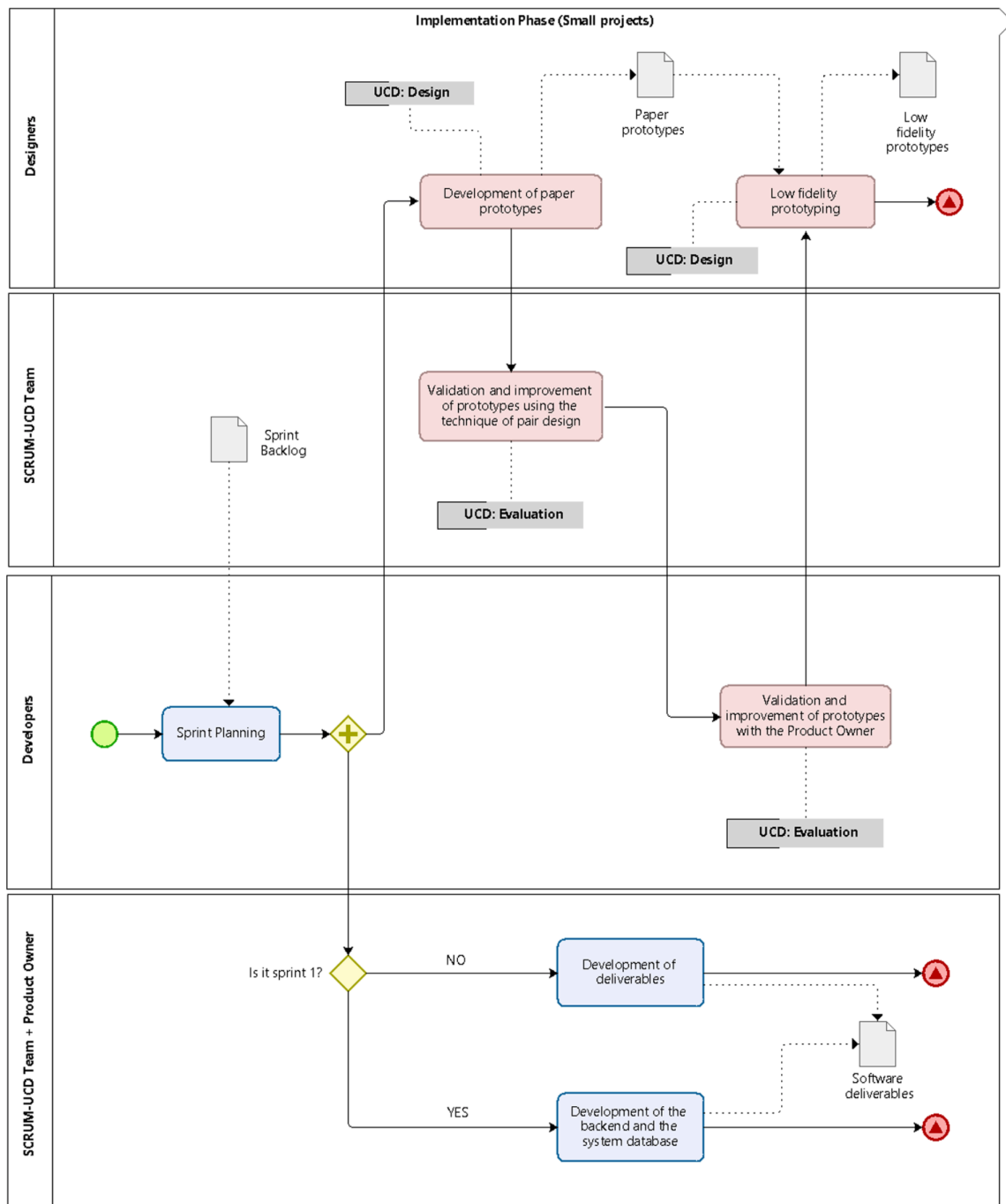


Figura 7 – Fase de implementação do projeto. Fonte: (ARGUMANIS; MOQUILLAZA; PAZ, 2021)

Na parte anterior do funil são realizadas as atividades que produzem uma visão geral do design conceitual e, após o início das sprints, na parte posterior do funil, são realizadas as atividades de design individual para cada funcionalidade. É também na parte posterior que ocorre a sincronização das atividades de UX com as atividades de desenvolvimento.

A parte anterior do funil é bastante semelhante com a *sprint 0* do framework Scrum-

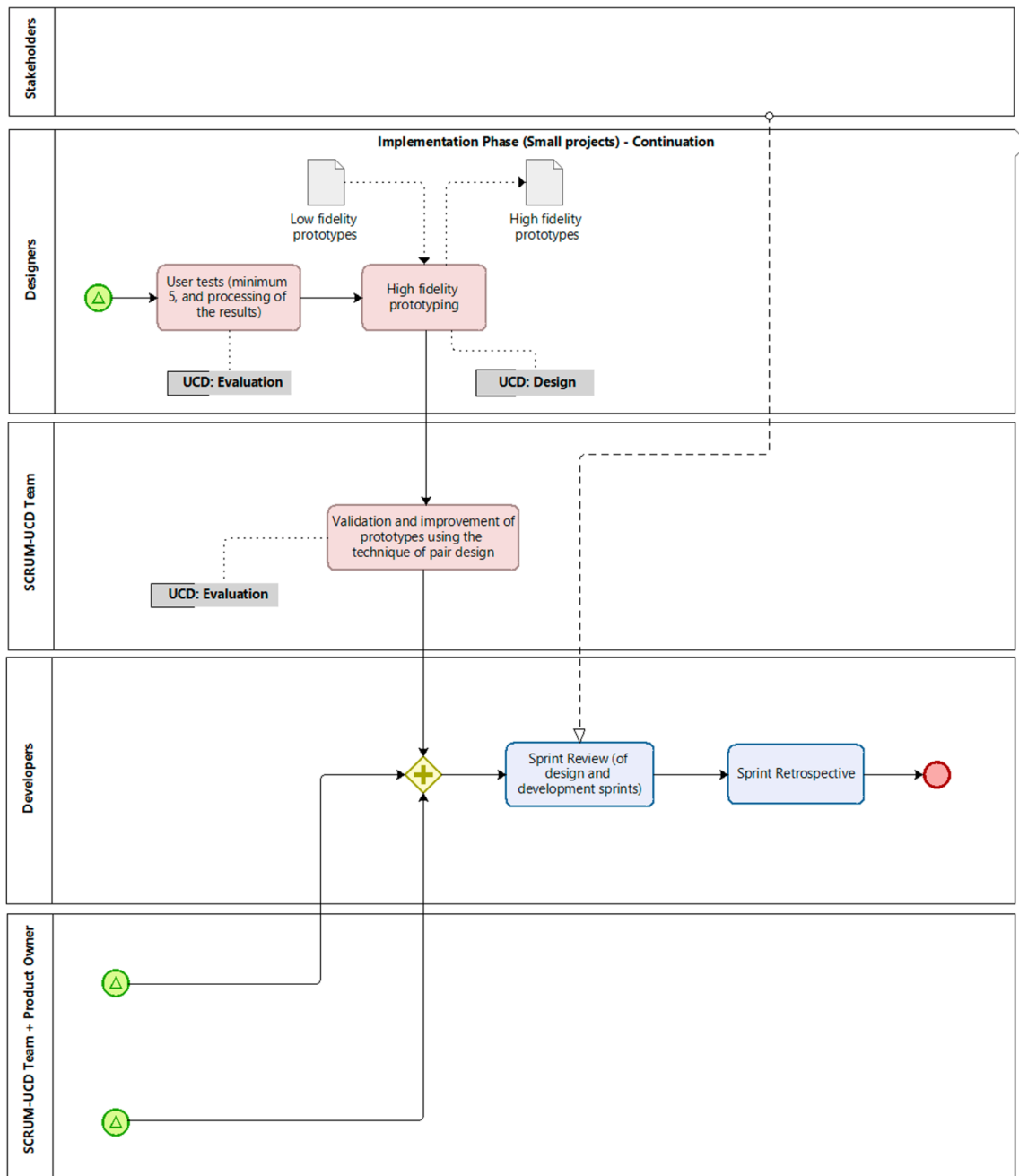


Figura 8 – Fase de implementação do projeto (continuação). Fonte: (ARGUMANIS; MOQUILLAZA; PAZ, 2021)

UCD (ARGUMANIS; MOQUILLAZA; PAZ, 2021) que contempla as fases de Iniciação e Planejamento; e a parte posterior do funil é equivalente à fase de Implementação. Trata-se da fase de entendimento da ecologia e necessidades dos usuários para o estabelecimento do design conceitual. Argumanis, Moquillaza e Paz (2021) explicam que essa fase anterior ao desenvolvimento é necessária por causa da natureza da UX - o design da experiência não pode ser ajustado e refatorado da mesma forma como é feito com o código. O tempo reservado para o funil anterior irá depender do tamanho e da complexidade do sistema a ser desenvolvido. Como resultado dessas atividades serão produzidos:

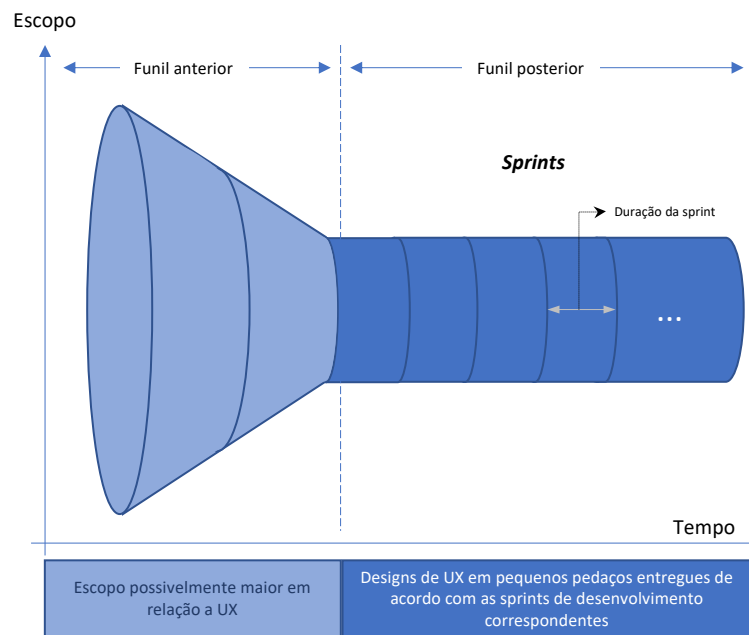


Figura 9 – Modelo funil de UX ágil. Adaptado de (HARTSON; PYLA, 2018)

- Uma visão geral do produto, o esqueleto onde serão colocadas as funcionalidades;
- Um design conceitual sólido para guiar o design das funcionalidades;
- Um design inicial *top-down*.

Na parte posterior do funil, assim como na fase de implementação do framework Scrum-UCD (ARGUMANIS; MOQUILLAZA; PAZ, 2021), as sprints de design ocorrem em paralelo às sprints de desenvolvimento, sendo sincronizadas de tal forma que enquanto os desenvolvedores implementam a sprint n , os designers estão trabalhando no design da sprint $n + 1$. Cada sprint de design contém um mini ciclo de UX contendo as atividades de análise, design, prototipação e avaliação (HARTSON; PYLA, 2018). Para Hartson e Pyla (2018), todas as atividades da sprint, seja de design ou desenvolvimento, são realizadas para desenvolver-se um escopo pequeno - uma única funcionalidade (*feature*) do produto ou sistema.

A história do usuário é o elemento que conecta a UX com a Engenharia de Software (HARTSON; PYLA, 2018). Na parte posterior do funil de UX ágil, o designer de UX apoia o responsável pela escrita das histórias de usuário e auxilia em sua priorização para as sprints.

4.3 Abordagem Combinada de Desenvolvimento Ágil, UCD e Lean Startup

Zorzetti et al. (2020) apresentam um modelo de desenvolvimento de software que contempla desenvolvimento ágil, *Lean Startup* e design centrado no usuário (UCD). O modelo foi gerado a partir de um mapeamento do fluxo de trabalho de dois times de desenvolvimento em uma grande multinacional de TI, que receberam treinamento e passaram a utilizar as três abordagens de forma combinada.

O participantes dos times de desenvolvimento atuam em três papéis:

- Designer do Produto (*Product Designer*): facilitador que viabiliza a comunicação com os usuários;
- Gerente do Produto (*Product Manager*): pessoa que representa a visão do negócio;
- Engenheiro de Software: desenvolvedor de software responsável por implementar a solução, participando também de atividades como entrevistas dos usuários.

O processo é dividido em três fases sequenciais: (1) Definição do Escopo (*Scoping*), (2) Descoberta e Modelagem (*Discovery and Framing*) e (3) Iteração (*Iteration*).

A fase de Definição do Escopo, apresentada na Figura 10, envolve atividades dirigidas a entender o problema a ser resolvido e ter uma visão geral do escopo de trabalho. Algumas das técnicas utilizadas nessa fase são: *Brainstorming*, para identificar abordagens de solução, e *Hopes and Fears*, para revelar expectativas e dúvidas sobre o projeto (ZORZETTI et al., 2020).

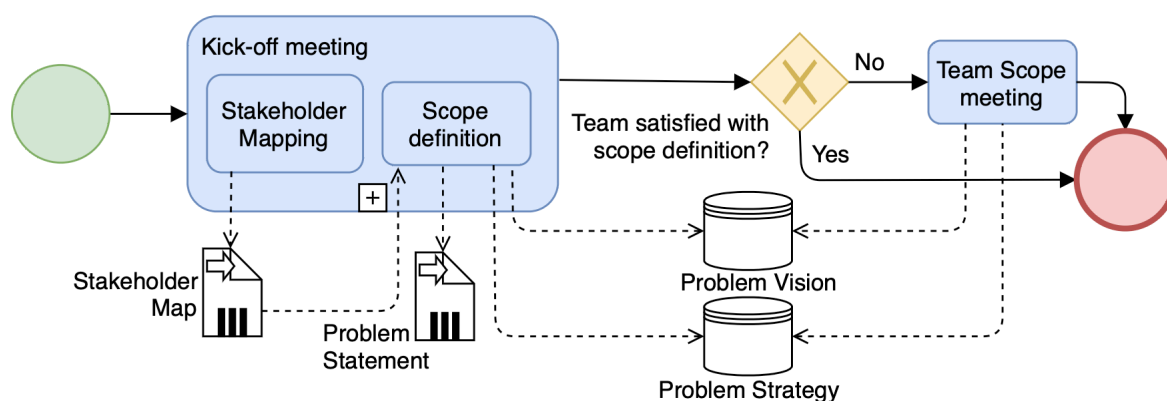


Figura 10 – Fase de Definição do Escopo (Scoping Phase). Fonte: (ZORZETTI et al., 2020)

A fase seguinte é dividida em duas etapas: Descoberta e Modelagem do Produto, apresentadas nas Figuras 11 e 12. O principal papel dessa fase é o Designer do Produto, responsável por guiar o time no correto entendimento das necessidades do usuário.

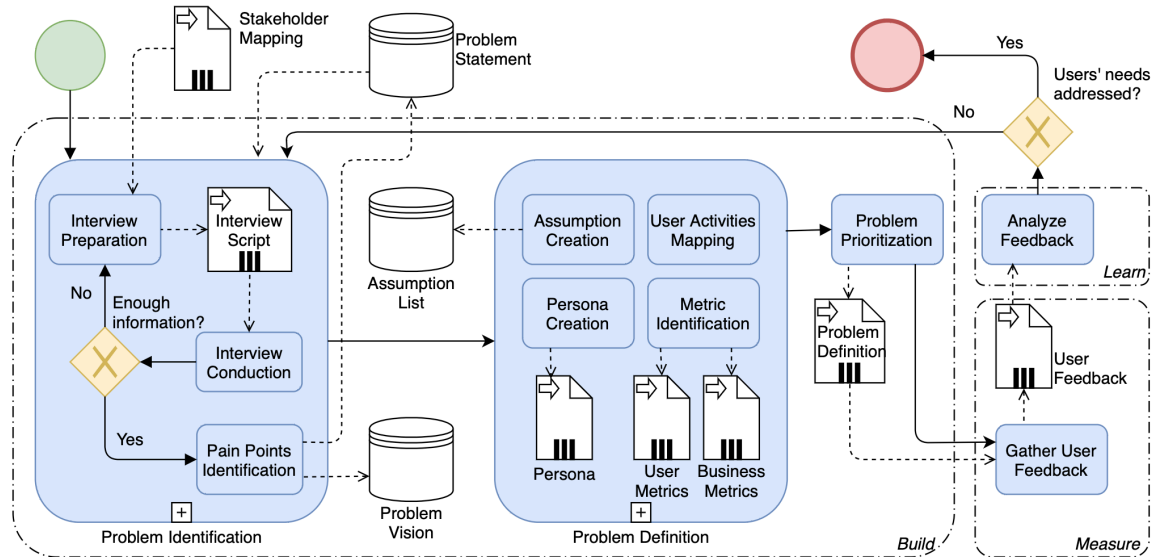


Figura 11 – Etapa de Descoberta (*Discovery Stage*). Fonte: (ZORZETTI et al., 2020)

A etapa de Descoberta visa aprofundar o entendimento do problema aplicando uma série de técnicas, tais como *Interview Preparation*, *Affinity Mapping*, *Topic Mapping*, *Brainstorming*, *User Flow*, *Service Blueprint*, *Journey Mapping*, *Two by Two Matrix*, *Now-near-next* e pesquisa etnográfica (ZORZETTI et al., 2020).

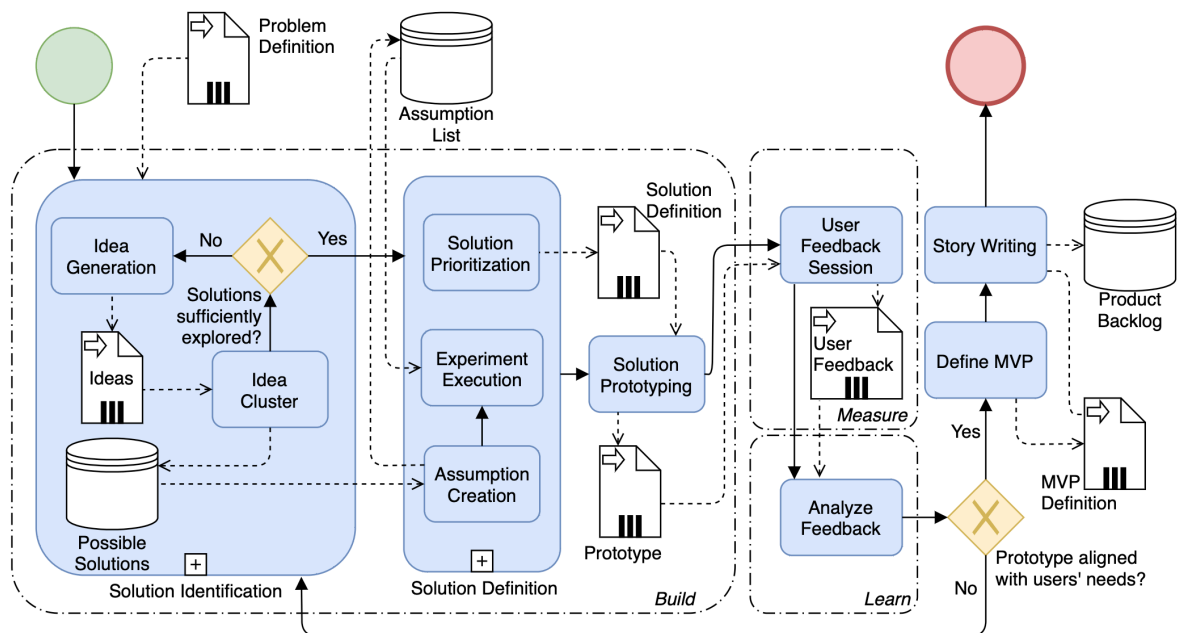


Figura 12 – Etapa de Modelagem do Produto (*Framing Stage*). Fonte: (ZORZETTI et al., 2020)

A etapa de Modelagem do Produto tem o objetivo de identificar múltiplas soluções para o problema e ao final escolher a mais adequada que irá definir o produto mínimo viável ou MVP, do inglês *Minimum Viable Product*. Para isso, além de algumas das técnicas já citadas, também são criados protótipos usando as técnicas de *wireframe* e *mockup* (ZORZETTI et al., 2020).

A fase de Iteração é apresentada na Figura 13. É a fase em que o produto será desenvolvido em pequenos ciclos até que todas as necessidades do negócio e dos usuários sejam atendidas. O time tem reuniões semanais com os stakeholders, além de reuniões para promover a troca de informações como o *Office Standup* e as *Tech Talks* (ZORZETTI et al., 2020).

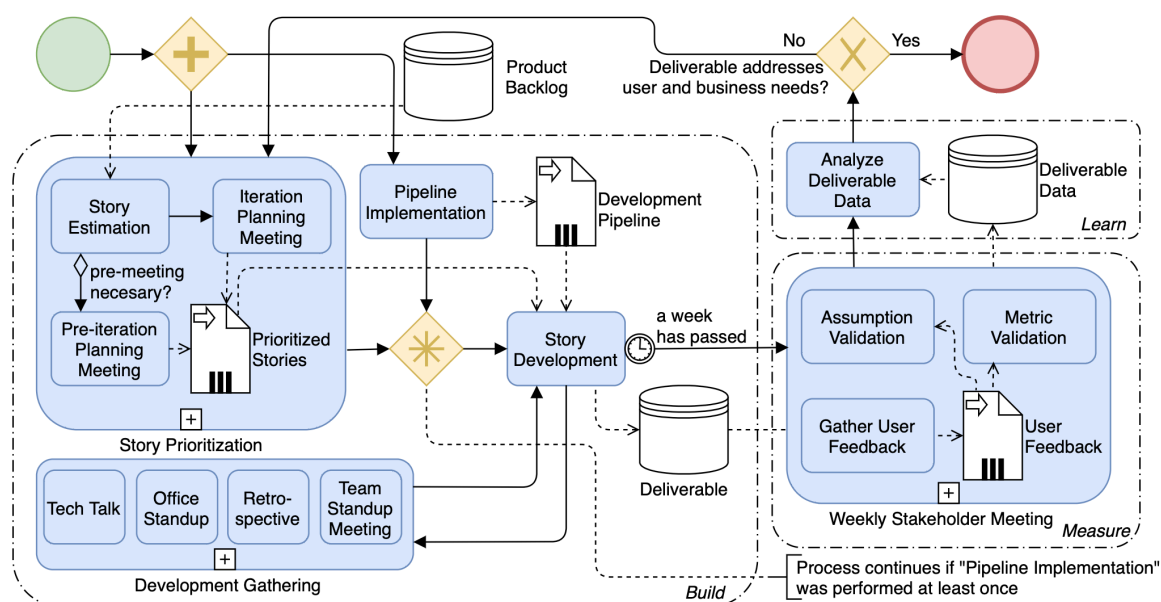


Figura 13 – Fase de Iteração (*Iteration Phase*). Fonte: (ZORZETTI et al., 2020)

4.4 User-Centered Behavioral (UCB) Software Development Model

Güncan e Durdu (2021) buscam integrar as disciplinas de ES e Interação Humano-Computador (HCI), propondo um modelo de desenvolvimento de software iterativo baseado na abordagem ágil de *behavior-driven development* (BDD) e técnicas de HCI.

Segundo os autores, o uso do BDD proporciona uma comunicação efetiva entre os stakeholders em todos os níveis, facilitando a integração dos usuários no processo de desenvolvimento de software. O ciclo de implementação do BDD usa cenários que são escritos em uma linguagem de domínio específico, muito próxima da linguagem natural, usando templates com palavras-chave. Dessa forma, os cenários podem ser entendidos por

todos os envolvidos no projeto e são facilmente automatizados e utilizados para realizar os testes de aceitação (GüNCAN; DURDU, 2021).

O modelo UCB é organizado em quatro fases. As fases de Análise e Design, Implementação e Avaliação são executadas em todas as iterações, enquanto a fase de Implantação ocorre apenas quando uma release é entregue ao cliente. Além disso, uma etapa inicial, chamada Iteração 0, é conduzida para realizar uma análise geral do sistema a ser desenvolvido. O modelo de desenvolvimento de software UCB é ilustrado na Figura 14.

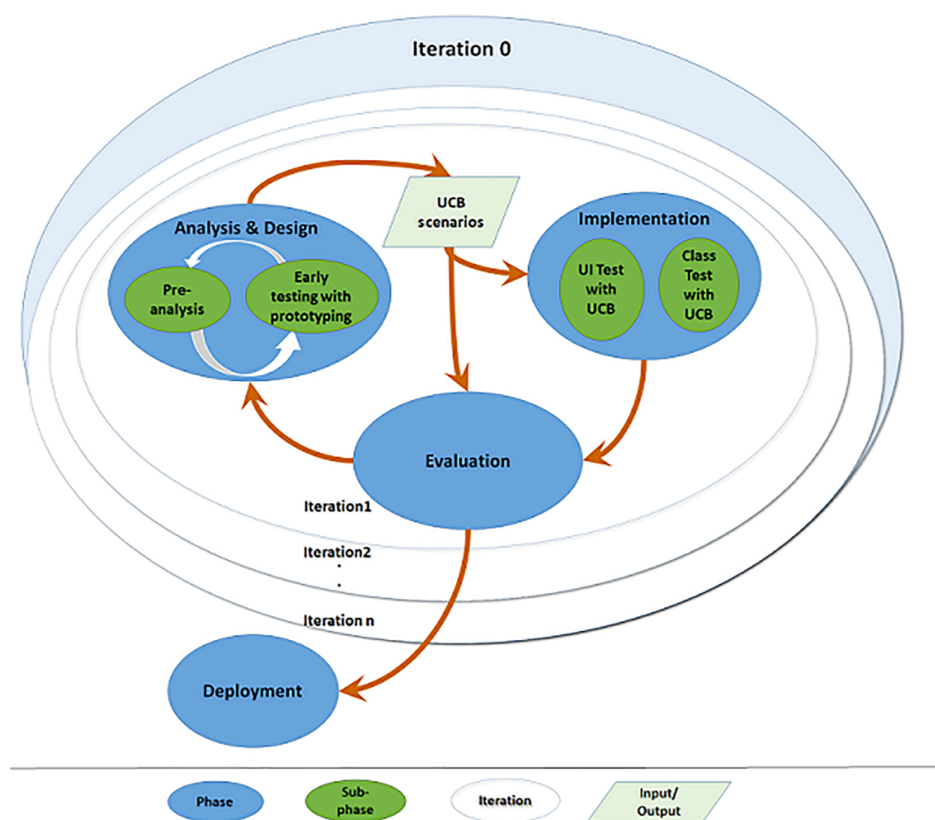


Figura 14 – Modelo de desenvolvimento de software UCB. Fonte: (GüNCAN; DURDU, 2021)

Na Iteração 0, as necessidades e prioridades são identificadas e os requisitos de alto nível são estabelecidos. A arquitetura de software começa a ser desenhada, preparando o projeto para o início do desenvolvimento na Iteração 1. Destaca-se nesta fase a elaboração dos guias de estilo pelos designers de UX, que irão apoiar o desenvolvimento consistente dos protótipos nas iterações seguintes (GüNCAN; DURDU, 2021).

As iterações seguintes executam o ciclo de Análise & Design, Implementação e Avaliação, em ciclos de duas a três semanas. Na fase de Análise & Design, analistas trabalham com clientes e usuários para quebrar os requisitos de alto nível em histórias de usuário e transformar o comportamento das histórias em cenários UCB. Estas informações serão então utilizadas para a elaboração dos protótipos (GüNCAN; DURDU, 2021).

Na fase de Implementação, os cenários UCB são implementados e os testes automatizados dos cenários são executados. Na fase de avaliação, os cenários que passaram nos testes são adicionados a um *buffer* de cenários prontos para implantação. Os cenários do *buffer* são então submetidos à avaliação de usabilidade com especialistas e usuários finais. Por fim, na fase de Implantação, quando um conjunto de requisitos do cliente estão prontos, sistema é liberado para utilização (GüNCAN; DURDU, 2021).

O modelo de desenvolvimento UCB prevê a atuação de cinco papéis (GüNCAN; DURDU, 2021):

- Clientes e usuários: participam em quase todas as fases, fornecendo as informações e testando protótipos e o produto final;
- Analistas: trabalham colaborativamente com clientes e usuários para identificar os requisitos e posteriormente transformá-los em cenários UCB;
- Testadores: grupo que inclui usuários e clientes trabalhando ao lado de engenheiros de teste e especialistas em UX para testar cenários, protótipos e o produto de cada release;
- Especialista UX: responsável pela criação do guia de estilo na Iteração 0 e dos protótipos ao longo das demais iterações;
- Desenvolvedores: fornecem o feedback técnico na Iteração 0 e implementam os cenários ao longo das iterações seguintes.

As principais contribuições do modelo, segundo os autores, ocorrem a partir do uso dos cenários UCB, que facilitam a comunicação entre todos os envolvidos no projeto, oferecem uma especificação detalhada dos requisitos de software, facilitam o desenvolvimento de protótipos e o teste do software implementado (GüNCAN; DURDU, 2021).

5 Integração Engenharia de Requisitos e Experiência do Usuário no GPS

Este capítulo apresenta uma discussão sobre o sobreposição entre as atividades da Engenharia de Requisitos e da Experiência do Usuário (nominada pelos membros como REUX) no contexto do Guia para o Desenvolvimento de Produtos e Soluções (GPS), desenvolvido no âmbito do Projeto *Transformação Digital de Serviços Públicos do Governo Brasileiro*, junto a Secretaria de Governo Digital (SGD) do Ministério da Economia (ME). Esse Projeto de Pesquisa e Desenvolvimento é uma parceria ME e UnB, mais especificamente entre a SGD e pesquisadores do laboratório ITRAC da FGA (UnB).

5.1 O Guia para o Desenvolvimento de Produtos e Soluções

O Guia para o Desenvolvimento de Produtos e Soluções (GPS) é um guia para o desenvolvimento de produtos e soluções, criado com base em princípios, valores, práticas e técnicas ágeis, as quais devem ser adaptadas à realidade do Governo Federal. Seu intuito é servir como guia e orientador para a atuação prática das Startups voltadas aos Projetos de Transformação Digital. O GPS é organizado em três fases principais: Diagnóstico, Execução e Encerramento, conforme ilustrado na Figura 15.



Figura 15 – Visão Geral do GPS: Fonte: <http://itrac.unb.br/gps/>

A fase de Diagnóstico tem como objetivo estabelecer os objetivos e produtos do projeto (Visão do Projeto), e formar a Startup.

A fase de Execução contempla as atividades de Engenharia de Software necessárias para a construção do produto ou solução. Essa fase é realizada de maneira iterativa e

incremental, com base na proposta do SCRUM, visando otimizar a previsibilidade e controlar o risco. Ao final de cada ciclo é realizada uma entrega compondo um incremento do produto. Uma ou mais entregas compõem uma RELEASE, a qual é disponibilizada em ambiente de produção para utilização do usuário final. A fase de Execução está organizada nas seguintes macro-atividades:

- Planejamento do Ciclo
- Execução do Ciclo
- Avaliação Contínua da Execução do Ciclo
- Realização das Entregas do Ciclo
- Avaliação das Entregas do Ciclo
- Liberação das Entregas do Ciclo

A fase de Encerramento refere-se à finalização dos trabalhos acordados entre Ministério da Economia (Secretaria de Governo Digital) e Órgãos parceiros. Neste momento ocorre o encerramento formal do projeto, da startup e do produto/solução, sendo que esse último é repassado para o órgão que irá mantê-lo.

5.2 Análise da Integração REUX no contexto do GPS

As atividades de Engenharia de Requisitos e de Experiência do Usuário ocorrem idealmente em todas as fases e macro-atividades do GPS. A Figura 16 ilustra como estas atividades são realizadas de forma paralela e integrada ao longo do ciclo de vida de um projeto.

A primeira coluna da Figura 16, *Engenharia de Requisitos*, apresenta as atividades de requisitos que em geral podem ser realizadas em cada uma das fases. A terceira coluna, *User Experience*, identifica as atividades mais comuns de UX que podem ser executadas. A coluna do meio, *Integração*, descreve as atividades que são realizadas em conjunto e que dizem respeito a ambas as áreas. As linhas pontilhadas ligam as atividades que, embora às vezes com nomes diferentes, geram os mesmos resultados.

Evidencia-se, dessa forma, que a Engenharia de Requisitos e a Experiência de Usuário possuem um grande sobreposição de atividades ao longo de todo o processo de desenvolvimento do projeto. Corroborando com os trabalhos apresentados nos capítulos anteriores, fica ressaltada a importância de que os profissionais envolvidos nas atividades dessas duas áreas trabalhem de forma colaborativa para o desenvolvimento de um projeto

que atenda às necessidades dos usuários do produto, como também de todos os envolvidos no projeto.

		INTEGRAÇÃO REUX		
		Atividades		
		Gestão da qualidade Comunicação estratégica		
		ENGENHARIA DE REQUISITOS	INTEGRAÇÃO	USER EXPERIENCE
DIAGNÓSTICO		<ul style="list-style-type: none"> - Identificar objetivos de Negócio. - Identificar e Propor solução. - Estabelecer os objetivos do Produto. - Identificar RFs e RNFs (alto nível). - Identificar Restrições e Condições para o Produto; - Estabelecer backlog do Produto (MVPs). - Alinhar Objetivos de Negócio x Produto x Resultados Esperados (valor). 	<ul style="list-style-type: none"> - Levantar antecedentes, materiais, histórico do tema, pesquisas, dados e outras fontes de entrada. - Identificar usabilidade comportamental (eficácia e eficiência) e usabilidade emocional (grau em que um produto é desejável ou serve a uma necessidade além do objetivo funcional tradicional). - Definir, implementar e gerenciar recursos da experiência do usuário, componentes e plataformas de produtos. 	<ul style="list-style-type: none"> - Validação da demanda e alinhamento das expectativas, - Benchmarking, - Mapeamento e validação da proposição de valor. - Estudos etnográficos. - Definição de oportunidades de inovação.
		- Identificar Problema e Necessidades.		- Estruturação do problema.
		- Identificar os stakeholders		- Identificação dos perfis de usuários; - Entendimento da diversidade de usuários.
PLANEJAMENTO DO CICLO		- Estabelecer a Estratégia de Gerenciamento do Backlog (refinamento, rastreabilidade, etc.)		
		- Definir conceito de pronto		- Geração de conceitos.
		- Priorizar o backlog da Sprint	- Definição de conceitos.	<ul style="list-style-type: none"> - Verificação dos objetivos a serem alcançados no final do ciclo. - Definição de práticas e metodologias a serem utilizadas no ciclo; - Entendimento dos contextos específicos de forma aprofundada. - Analisar e sintetizar as ideias. - Definição de métricas de sucesso.
EXECUÇÃO DO CICLO		- Garantir a comunicação dos requisitos entre as partes envolvidas (equipe técnica e negocio). - Refinar o Backlog da Sprint.		- Estruturação da Arquitetura da Informação. - Definição do fluxo/jornadas - Guia de estilo.
		- Estabelecer regras de negócio e critérios de aceitação.		- Entendendo a interação com o produto / serviço a partir dos diferentes usuários.
		- Manter atualizado o backlog da sprint.	- Gestão da execução do ciclo. - Foco nos requisitos do sistema e dos usuários.	<ul style="list-style-type: none"> - Desenhando ideias. - Wireframes. - Especificação e Validação de UI. - Validação de taxonomia. - Especificando o design a partir dos feedbacks de usuários.
REALIZAÇÃO DAS ENTREGAS DO CICLO		- Comunicar o encerramento da execução backlog da Sprint.		- Prototipação e teste e Tangibilização de Soluções. - Avaliação de acessibilidade. - Testes de usabilidade para validação.
		- Garantir a utilização dos requisitos (RF e RNFs) e critérios de aceitação para a validação das entregas	← - Consolidação de requisitos. →	- Satisfação do usuário com o produto - Avaliação de impacto - pos acompanhamento;
AVALIAÇÃO DOS RESULTADOS DO CICLO		- Garantir o cumprimento do conceito de pronto;	- Garantir adequação e utilização de requisitos	- Garantir a viabilidade da solução;
		- Garantir o cumprimento do conceito de pronto;	- Foco no usuário e no alinhamento na estratégia de negócio	- Identificação de pontos de melhoria no processo de execução do ciclo; - Avaliação de métricas de sucesso;
LIBERAÇÃO DAS ENTREGAS DO CICLO		- Atualizar o backlog do produto; - Atualizar o alinhamento entre o backlog do produto, objetivo de negócio e resultados esperados. - Garantir o cumprimento dos RNFs para a liberação das entregas	- Garantir a viabilidade dos requisitos adequados aos usuários, à tecnologia e aos negócios; - Finalização do ciclo;	- Adequação das entregas aos formatos necessários;
		- Atualizar o backlog do produto; - Atualizar o alinhamento entre o backlog do produto, objetivo de negócio e resultados esperados. - Garantir o cumprimento dos RNFs para a liberação das entregas	- Compreensão de feedback dos usuários e devolutiva à equipe de desenvolvimento;	- Revisão de funções a partir das interações do usuário; - Atualização do Guia de estilo;
FIM		- Encerrar o backlog do Produto;		- Comunicação estratégica, da solução, para o público;

Figura 16 – Integração entre as atividades de Requisitos e Experiência do Usuário no GPS.
Fonte: elaborada pelos autores

6 Considerações Finais

Este relatório apresentou conceitos e perspectivas relacionadas à incorporação da UX nos processos de desenvolvimento de software. Esse é um movimento relativamente recente, conforme evidenciado pelas datas de publicação dos trabalhos analisados. A expectativa dos benefícios trazidos pela aplicação dos métodos e das técnicas de UX no desenvolvimento de software tem atraído interesse tanto da indústria quanto da academia para a interseção entre essas duas áreas.

Considerando o sobreposição percebido entre as atividades do designer de UX e algumas das atividades da ER, o relatório apresentou perspectivas de como a ER vê a UX nos processos de requisitos de software. Observou-se que enquanto a dimensão da usabilidade encontra-se amadurecida no contexto do desenvolvimento de software, a incorporação dos aspectos hedônicos da UX ainda é um desafio para a ER. Já o framework de integração entre essas áreas apresentou que o design de uma boa UX requer a colaboração dos profissionais de ER e UX ao longo de todo o processo de desenvolvimento de software.

O relatório abordou também quatro propostas de integração do design de UX no desenvolvimento ágil de software. Os modelos apresentados ressaltam a necessidade de iniciar as atividades de UX nas fases iniciais do desenvolvimento de software. Outro ponto em comum nas propostas de integração analisadas é a divisão das atividades de UX em dois estágios distintos, um conjunto de atividades de mais alto nível realizadas antes do início das atividades de implementação (sprint 0) e outro conjunto de atividades que são desenvolvidas iterativamente e paralelamente aos ciclos de desenvolvimento, dedicadas à prototipação e à avaliação dos entregáveis produzidos.

Para além dos aspectos em comum, cada proposta mostrou particularidades com relação aos papéis envolvidos, mas principalmente em relação a métodos e técnicas aplicadas, sejam do universo dos métodos ágeis ou do amplo arcabouço da área de UX. Tais diferenças refletem possivelmente os contextos particulares em que cada modelo foi concebido.

Finalmente, o sobreposição das atividades de ER e UX analisado no contexto do GPS mostra-se consoante com as propostas de integração do design de UX no desenvolvimento ágil de software. Desde a introdução de atividades de UX já na fase de Diagnóstico, em paralelo às definições iniciais do produto, até o encerramento do projeto com a compreensão do feedback dos usuários essa colaboração entre as áreas deve ser almejada de forma a produzir soluções de software bem sucedidas.

Esperamos que as perspectivas e propostas apresentadas neste relatório possam

servir para aprimorar o entendimento acerca da aplicação do design de UX no desenvolvimento de software, bem como inspirar a todos na criação de produtos de software cada vez mais efetivos e impactantes para os usuários finais.

Referências

ANITHA, P.; PRABHU, B. Integrating requirements engineering and user experience design in Product life cycle Management. In: *2012 First International Workshop on Usability and Accessibility Focused Requirements Engineering (UsARE)*. Zurich, Switzerland: IEEE, 2012. p. 12–17. ISBN 978-1-4673-1846-4 978-1-4673-1845-7. Disponível em: <<http://ieeexplore.ieee.org/document/6226784/>>. Citado 4 vezes nas páginas 7, 8, 10 e 11.

ARGUMANIS, D.; MOQUILLAZA, A.; PAZ, F. A Framework Based on UCD and Scrum for the Software Development Process. In: SOARES, M. M.; ROSENZWEIG, E.; MARCUS, A. (Ed.). *Design, User Experience, and Usability: UX Research and Design*. Cham: Springer International Publishing, 2021. v. 12779, p. 15–33. ISBN 978-3-030-78220-7 978-3-030-78221-4. Series Title: Lecture Notes in Computer Science. Disponível em: <https://link.springer.com/10.1007/978-3-030-78221-4_2>. Citado 7 vezes nas páginas 13, 14, 15, 16, 17, 18 e 19.

BENYON, D. *Designing User Experience: A Guide to Hci, UX and Interaction Design*. 4th ed. edição. ed. Harlow, United Kingdom: Pearson, 2019. ISBN 978-1-292-15551-7. Citado na página 6.

GÜNCAN, D.; DURDU, P. O. A user-centered behavioral software development model. *Journal of Software: Evolution and Process*, v. 33, n. 2, fev. 2021. ISSN 2047-7473, 2047-7481. Disponível em: <<https://onlinelibrary.wiley.com/doi/10.1002/smr.2274>>. Citado 3 vezes nas páginas 22, 23 e 24.

HARTSON, R.; PYLA, P. *The UX Book: Agile UX Design for a Quality User Experience*. 2nd edition. ed. Cambridge, MA: Morgan Kaufmann, 2018. ISBN 978-0-12-805342-3. Citado 4 vezes nas páginas 7, 9, 16 e 19.

KASHFI, P. et al. A Conceptual UX-Aware Model of Requirements. In: BOGDAN, C. et al. (Ed.). *Human-Centered and Error-Resilient Systems Development*. Cham: Springer International Publishing, 2016. v. 9856, p. 234–245. ISBN 978-3-319-44901-2 978-3-319-44902-9. Series Title: Lecture Notes in Computer Science. Disponível em: <http://link.springer.com/10.1007/978-3-319-44902-9_15>. Citado 3 vezes nas páginas 8, 9 e 10.

KHARSHILADZE, I.; LUO, Q. *How Requirements Elicitation Process Takes User Experience(UX) Into Account*. Tese (Bachelor of Science Thesis) — University of Gothenburg, Göteborg, Sweden, fev. 2015. Citado 3 vezes nas páginas 8, 9 e 10.

MIRNIG, A. G. et al. A Formal Analysis of the ISO 9241-210 Definition of User Experience. In: *Proceedings of the 33rd Annual ACM Conference Extended Abstracts on Human Factors in Computing Systems*. New York, NY, USA: Association for Computing

- Machinery, 2015. (CHI EA '15), p. 437–450. ISBN 978-1-4503-3146-3. Disponível em: <<https://doi.org/10.1145/2702613.2732511>>. Citado na página 6.
- OHASHI, K. et al. Focusing requirements elicitation by using a UX measurement method. In: *2018 IEEE 26th international requirements engineering conference (RE)*. IEEE, 2018. p. 347–357. ISBN 978-1-5386-7418-5. Place: Banff, AB, Canada tex.eventtitle: 2018 IEEE 26th international requirements engineering conference (RE). Disponível em: <<https://ieeexplore.ieee.org/document/8491149/>>. Citado 3 vezes nas páginas 8, 10 e 11.
- STILL, B.; CRANE, K. *Fundamentals of User-Centered Design: A Practical Approach*. Boca Raton: CRC Press, 2017. ISBN 978-1-315-20092-7. Citado na página 7.
- SUTCLIFFE, A. User-Oriented Requirements Engineering. In: EBERT, A. et al. (Ed.). *Usability- and Accessibility-Focused Requirements Engineering*. Cham: Springer International Publishing, 2016. v. 9312, p. 11–33. ISBN 978-3-319-45915-8 978-3-319-45916-5. Series Title: Lecture Notes in Computer Science. Disponível em: <http://link.springer.com/10.1007/978-3-319-45916-5_2>. Citado 2 vezes nas páginas 8 e 11.
- TOSI, F. From User-Centred Design to Human-Centred Design and the User Experience. In: *Design for Ergonomics*. Cham: Springer International Publishing, 2020. v. 2, p. 47–59. ISBN 978-3-030-33561-8 978-3-030-33562-5. Series Title: Springer Series in Design and Innovation. Disponível em: <http://link.springer.com/10.1007/978-3-030-33562-5_3>. Citado 2 vezes nas páginas 6 e 7.
- ZORZETTI, M. et al. An Empirical-informed Work Process Model for a Combined Approach of Agile, User-Centered Design, and Lean Startup. In: *19th Brazilian Symposium on Software Quality*. São Luís Brazil: ACM, 2020. p. 1–10. ISBN 978-1-4503-8923-5. Disponível em: <<https://dl.acm.org/doi/10.1145/3439961.3439967>>. Citado 3 vezes nas páginas 20, 21 e 22.