



UNIVERSIDADE FEDERAL DE PERNAMBUCO
CENTRO DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

SILVIO CAETANO DE SÁ

O Futuro dos Métodos de Desenvolvimento de Software

RECIFE

2019

SILVIO CAETANO DE SÁ

O Futuro dos Métodos de Desenvolvimento de Software

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de Pernambuco, como requisito parcial para a obtenção do título de mestre em Ciência da Computação.

Orientador: Prof. Dr. Hermano Perrelli de Moura

RECIFE

2019

Ficha de identificação da obra elaborada pelo autor,
através do programa de geração automática do SIB/UFPE

Sá, Silvio Caetano de .
O futuro dos métodos de desenvolvimento de software / Silvio Caetano de
Sá. - Recife, 2019.
101 p., tab.

Orientador(a): Hermano Perrelli de Moura
Dissertação (Mestrado) - Universidade Federal de Pernambuco, Centro de
Informática, Programa de Pós-Graduação em Ciência da Computação, 2019.
Inclui referências, apêndices.

1. Engenharia de Software. 2. Futuro do software. 3. Delphi. 4. Cenários.
I. Moura, Hermano Perrelli de. (Orientação). II. Título.

000 CDD (22.ed.)

Silvio Caetano de Sá

O Futuro dos Métodos de Desenvolvimento de Software

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de Pernambuco, como requisito parcial para a obtenção do título de mestre em Ciência da Computação.

Aprovado em: 12/09/2019

BANCA EXAMINADORA

Prof. Dr. Hermano Perrelli de Moura
Centro de Informática / UFPE
(Orientador)

Prof. Dr. Cristiano Coelho de Araújo
Centro de Informática / UFPE

Prof. Dr. Alixandre Thiago Ferreira de Santana
Faculdade de Ciências da Administração de Garanhuns / UFRPE

AGRADECIMENTOS

Gratidão é tempo de parar e de pensar, é momento de reflexão,

Me lembro de tantos momentos, de alguma tristeza e de muita paixão.

A Deus devo tudo, minha vida, saúde, paz e alegria,

Não deixo de falar e agradecer a ele, toda noite, todo dia.

Se vou falar em gratidão, vem logo a minha mente a figura de meus pais,

Sempre estiveram do meu lado, não me deixaram jamais.

Não me esqueço do que fizeram, de todo amor que recebi,

Do carinho e do afeto, do exemplo que neles vi.

Penso tanto sobre isso, mas não sei nem o que dizer,

Devo a eles minha vida, só tenho que me inclinar e agradecer.

Um dia deixei meus pais, pra morar noutro lugar,

Me arrumei com minha esposa, e fui com ela morar.

Eu nem percebi direito, muito tempo se passou,

Mas uma coisa eu tenho certeza, ela é o meu amor.

Minhas filhas lindas, minha neta, são minha inspiração,

Dei tudo que pude a elas, saúde, amor, exemplo e educação.

Tanta gente me ajudou, Prof. Hermano, colegas do GP2,

A banca que me examinou, e tudo que me propôs.

Não quero esquecer ninguém, pra não ficar encrencado,

Por isso paro por aqui, a todos meu muito obrigado!

A sabedoria é a coisa principal; adquiere pois a sabedoria, emprega tudo o que possuis na aquisição de entendimento. Exalta-a, e ela te exaltará; e, abraçando-a tu, ela te honrará. Dará à tua cabeça um diadema de graça e uma coroa de glória te entregará.

(Provérbios 4:7-9)

RESUMO

O software vem dominando todas as áreas do conhecimento humano, e essa tendência tende a crescer cada dia mais. O software tem provocado o aparecimento de novas tecnologias, de novos negócios, de novas formas do ser humano se relacionar; o impacto é gigantesco e irreversível. Essa dependência, no entanto, traz algumas questões preocupantes. A história da Engenharia de Software tem mostrado que alguns problemas no processo de desenvolvimento de software que existiam há décadas, continuam a persistir, como o alto custo de manutenção, a baixa qualidade dos processos e produtos, previsões, planejamentos precários, entre outros. Diante desse quadro questionamos se o software que está sendo desenvolvido hoje é confiável, é seguro; e mais ainda, se o software que será desenvolvido no futuro, que irá requerer muito mais complexidade diante dos novos desafios tecnológicos, de novas necessidades, será adequado, seguro e confiável. O aumento da incerteza aumentou a importância de identificar as tendências futuras. Sendo assim nos antecipamos, identificando os principais fatores que poderão afetar os métodos de desenvolvimento de software no futuro. Utilizamos o método de pesquisa de futuro Delphi, buscando o consenso de especialistas com conhecimento em engenharia de software sobre quais fatores mais influenciarão os métodos de desenvolvimento de software no futuro, e que foram a base para a construção de quatro cenários, que poderão auxiliar as empresas que desenvolvem software a se planejarem adequadamente, pois a prospectiva estratégica facilita a adaptação rápida às principais mudanças, buscando garantir a produção de software com a melhor qualidade possível.

Palavras-chave: Engenharia de Software, futuro do software, Delphi, cenários.

ABSTRACT

Software has dominated every area of human knowledge, and this trend tends to grow more and more. Software has triggered the emergence of new technologies, new businesses, new ways for humans to relate; the impact is gigantic and irreversible. This dependence, however, brings with it some worrying questions. The history of Software Engineering has shown that some problems in the software development process that existed for decades continue to persist, such as high maintenance costs, poor quality of processes and products, forecasts, poor planning, among others. Given this scenario we question whether the software being developed today is reliable, safe; Moreover, if the software that will be developed in the future, which will require much more complexity in the face of new technology challenges, new needs, will be adequate, secure, reliable. Increasing uncertainty has increased the importance of identifying future trends. Therefore, we anticipate by identifying the key factors that may affect software development methods in the future. We use the Delphi future research method, seeking the consensus of experts with software engineering knowledge on which factors will most influence software development methods in the future, and which were the basis for building four scenarios, which can help software developers plan properly, as strategic foresight facilitates rapid adaptation to key changes to ensure the highest quality software production possible.

Keywords: Software Engineering, software future, Delphi, scenarios.

LISTA DE ABREVIATURAS E SIGLAS

ABNT	Associação Brasileira de Normas Técnicas
ACM	Association for Computing Machinery
ASD	Adaptive Software Development
DDD	Design Driven Development
DSDM	Dynamic Systems Development Method
FDD	Feature Driven Development
IA	Inteligência Artificial
ICSE	International Conference on Software Engineering
IEEE	Institute of Electrical and Electronic Engineers
MACTOR	Metodologia dos Atores, Objetivos e Relações de Força
PDTI	Plano Diretor de Tecnologia da Informação
PETI	Planejamento Estratégico de Tecnologia da Informação
SEBRAE	Serviço Brasileiro de Apoio às Micro e Pequenas Empresas
SOA	Service Oriented Architecture
SQA	Software Quality Assurance
SWEBOK	Software Engineering Body of Knowledge
SXSW	South by Southwest
TCU	Tribunal de Contas da União
TFA	Technology Future Analysis
XP	Extreme Programming

LISTA DE ILUSTRAÇÕES

Figura 1.	Ranking BrandZ™ 2019 das empresas de maior valor de mercado do mundo.....	18
Figura 2.	Crise de Software 2.0. Fonte: (FITZGERALD, 2012).....	21
Figura 3.	Fluxograma de aplicação do método Delphi.....	32
Figura 4.	Paradigma clássico do ciclo de vida do software. Fonte: (WANG, KING, 2000).	40
Figura 5.	Uma tipologia para atributos de qualidade do processo de engenharia de software (KROEGER et al, 2013).....	42
Figura 6.	Modelo Paralelo do Processo de Requisitos (HICKEY, DAVIS, 2004). 49	
Figura 7.	Diagrama do Método XP. Fonte: http://www.inf.ufpr.br/silvia/ES/SweES/pdf/AgeisAl.pdf	52
Figura 8.	Diagrama do Método Scrum. Fonte: http://www.inf.ufpr.br/silvia/ES/SweES/pdf/AgeisAl.pdf	52
Figura 9.	Etapas da pesquisa.....	61
Figura 10.	Resultado da pesquisa para o ano de 1999 utilizando o Google Acadêmico.....	62
Figura 11.	Página inicial da ferramenta Delphi Decision Aid.....	68
Figura 12.	Sessão de teste da ferramenta Delphi Decision Aid.....	69
Figura 13.	E-mail convite para aplicação da Ferramenta Delphi Decision Aid.	70
Figura 14.	Página inicial do Questionário aplicado para a Primeira rodada.	71
Figura 15.	Página inicial do Questionário aplicado para a Segunda rodada.	71
Figura 16.	Diagrama de influência 1 – Interfaces mais naturais.	81
Figura 17.	Diagrama de influência 2 – IoT – Internet das Coisas.	81
Figura 18.	Diagrama de influência 3 – Transparência das atividades.....	82
Figura 19.	Diagrama de influência 4 – Construção ágil.	82
Figura 20.	Espaço de classificação das forças motrizes - Interfaces mais naturais..	84
Figura 21.	Espaço de classificação das forças motrizes – Internet das Coisas.....	84
Figura 22.	Espaço de classificação das forças motrizes – Transparência das atividades.....	85
Figura 23.	Espaço de classificação das forças motrizes – Construção Ágil.....	85

Gráfico 1.	Incidências dos argumentos nos resumos dos artigos.	62
Gráfico 2.	Distribuição das respostas para as perguntas do questionário.....	77
Quadro 1.	Comparativo das características entre a Previsão e a Prospecção (SCHENATTO et al, 2011).....	25
Quadro 2.	Convergências e divergências dos Modelos de Planejamento de Cenários da Shell e da Escola Francesa (JUNIOR et al, 2010).....	37
Quadro 3.	Modelo híbrido de Planejamento de Cenários, baseado nos modelos da Shell e da Escola Francesa (Adaptado de JUNIOR et al 2010).	39
Quadro 4.	Estrutura de tomada de decisão em pesquisa (adaptado de WOHLIN & AURUM, 2014).....	56
Quadro 5.	Perfil dos especialistas que responderam ao questionário.	65
Quadro 6.	Perfil dos especialistas convidados para testar a ferramenta Delphi Decision Aid.....	69
Quadro 7.	Respostas consolidadas do Questionário Avaliativo.....	74
Quadro 8.	Variáveis chaves identificadas a partir das respostas dos questionários.	75
Quadro 9.	Totalização da pontuação para as variáveis chaves.	76
Quadro 10.	Demonstrativo da probabilidade de desenvolvimento e de impacto potencial das variáveis chaves.	76
Quadro 11.	Perfil dos profissionais que participaram da elaboração dos cenários.	80
Quadro 12.	Atividades da equipe de elaboração dos Cenários.	80

LISTA DE TABELAS

Tabela 1.	Ciclo de vida do software e componentes (MILLS, 1999)	41
Tabela 2.	Fatores da qualidade do software (ASHRAFI, 2002).	44
Tabela 3.	Práticas e atividades relacionadas (MILLS, 1999)	46
Tabela 4.	Comparativo das ferramentas para aplicação do Delphi online.	67
Tabela 5.	Pontuação das questões a partir do quantitativo de respostas do Questionário Avaliativo.	75

SUMÁRIO

1. INTRODUÇÃO	16
1.1. Contexto.....	16
1.2. A Crise de Software	19
1.3. Problema de Pesquisa	21
1.4. O Futuro.....	24
1.5. Objetivos.....	26
1.6. Organização da Dissertação.....	26
2. Referencial Teórico	28
2.1. Estudos do Futuro	28
2.1.1. Método Delphi.....	29
2.1.1.1. Delphi Online	33
2.1.2. Método de Construção de Cenários	35
2.2. Software	39
2.2.1. Ciclo de Vida do Software	40
2.2.2. Qualidade do Software	41
2.3. Engenharia de Software	44
2.3.1. Práticas de Engenharia de Software	45
2.3.2. Processos de Engenharia de Software.....	46
2.3.3. Engenharia de Software Baseada em Componentes	47
2.3.4. Gestão do Conhecimento em Engenharia de Software	47
2.3.5. Engenharia de Requisitos	48
2.4. Metodologias de Desenvolvimento de Software	49
2.4.1. Métodos de Desenvolvimento Ágeis.....	51
2.4.2. Métodos de Desenvolvimento Orientados a Serviços.....	53

2.5. Análise	54
3. Metodologia da Pesquisa.....	55
3.1. Fases da Pesquisa.....	55
Fase Estratégica	56
1. Resultado da Pesquisa	56
2. Lógica da Pesquisa	56
3. Propósito da Pesquisa	57
4. Abordagem da Pesquisa	57
Fase Tática	57
5. Processo de Pesquisa	57
6. Metodologia da Pesquisa.....	58
Fase Operacional.....	58
7. Método de Coleta de Dados	58
8. Método de Análise de Dados.....	59
4. Resultados da Pesquisa.....	60
4.1. Etapas da Pesquisa	61
4.1.1. Realizar Pesquisa ad-hoc.....	61
4.1.2. Selecionar Especialistas	64
4.1.3. Organizar Dados para Aplicação do Método	66
4.1.4. Obter a Opinião dos Especialistas	66
Escolha da Ferramenta	67
Conclusões sobre o Uso do Método Delphi	72
4.1.5. Identificar Tendências	73
4.1.6. Analisar e Interpretar Resultados	76
4.1.7. Propor Melhorias.....	79
4.1.8. Criar Cenários Futuros	79
Listar Padrões e Tendências.....	80

Elaborar Diagramas de Influência.....	81
Listar as Forças Motrizes	83
Classificar as Forças Motrizes.....	83
4.2. Discussão dos resultados	85
5. Conclusões e Trabalhos Futuros	87
5.1. Contribuições	87
5.2. Trabalhos Futuros	88
Referências	89
APÊNDICE A	94
APÊNDICE B.....	96
APÊNDICE C.....	97

1. INTRODUÇÃO

Este capítulo traz uma visão geral do trabalho e está estruturado com as seguintes seções:

- 1.1 Contexto: descreve o contexto, a motivação, e o problema de pesquisa a ser analisado e respondido.
- 1.2. A Crise de Software: apresenta uma análise histórica dos principais problemas enfrentados no desenvolvimento de software.
- 1.3 Problema de Pesquisa: aponta o direcionamento para o problema de pesquisa a ser analisado e respondido.
- 1.4 O Futuro: apresenta uma introdução sobre os métodos de previsão e prospecção e aponta a alternativa usada neste trabalho.
- 1.5 Objetivos: destaca o objetivo geral do trabalho e também os objetivos específicos que devem ser alcançados.
- 1.6 Organização da Dissertação: mostra como os capítulos da dissertação estão organizados.

1.1. Contexto

É inegável a influência e a dependência cada vez maior que o software exerce em nossa sociedade, e como vem afetando profundamente nossas vidas, os negócios, a manufatura, a agricultura, a saúde, estando presente em praticamente todos os campos da atividade humana, sendo hoje a tecnologia mais importante no cenário mundial e deverá ser cada vez mais. Além disso, não foi possível prever que o software se tornaria essa tecnologia, sendo indispensável para os negócios, para a ciência e para a vida pessoal, bem como que o software permitiria a criação de tantas novas tecnologias, tais como engenharia genética e nanotecnologia, e permitiria o aperfeiçoamento de outras tecnologias já existentes, como as telecomunicações, entre outras (CASALEA et al, 2016; PRESSMAN, MAXIM, 2015; KROEGER et al, 2013).

Durante os anos 1990 surgiram muitas especulações de que o software estava mudando, onde os ciclos de desenvolvimento de software “foram muito apertados,

impulsionados por telecomunicações e por avanços técnicos generalizados na entrega de software” (RAMESH et al, 2002), enquanto isso o software se tornou essencial para os negócios, onde a capacidade das organizações e de seus produtos, sistemas e serviços para competir, adaptar e sobreviver dependeu cada vez mais de software (BOEHM, 2005).

A produção de software teve um grande impulso com a popularização da internet, quando nos anos 1990 a internet deixou de ser uma ferramenta acadêmica para se tornar um dispositivo pessoal e comercial comum (RAMESH et al, 2002), e mudou o modo de comunicação entre clientes e fornecedores, bem como a maneira de colocar os produtos no mercado e a maneira de fornecer suporte aos clientes (CRNKOVIC, 2001). Começou a se popularizar uma ampla variedade de modelos de novos negócios e em constante mudança, relacionados ao fornecimento de software e aos serviços mediados por softwares resultantes da tecnologia da internet e do e-commerce (FINKELSTEIN, KRAMER, 2000).

FINKELSTEIN & KRAMER (2000) observaram que no início dos anos 2000, de maneira geral, menos sistemas de software estavam sendo construídos, ao invés disso estavam sendo construídos componentes genéricos para serem vendidos nos mercados, sendo selecionados e comprados, com o desenvolvimento sendo redirecionado na configuração e na interoperabilidade.

A influência do software no mundo atual é enorme e permeia quase todos os aspectos da sociedade moderna. O software deixou de ser considerado um produto altamente especializado para ser um produto comum e essencial. Essa mercantilização resultou na crescente pressão do setor de software para desenvolver e fornecer maiores volumes de produtos e serviços de alta qualidade dentro das restrições de custo e cronograma que são mais rigorosas do que nunca (KROEGER et al, 2013).

O software passou a fazer parte dos negócios de modo irreversível e de forma cada vez mais intensa, de tal forma que os modelos de negócios mudaram completamente, sob forte influência do software. Chegamos ao ponto em que entre as empresas de maior valor de mercado do mundo, a maior parte são empresas de software, segundo o Ranking BrandZ™ 2019 (Figura 1), sendo a Amazon a mais valiosa, superando a Apple, que foi a mais valiosa nos últimos anos, e que produz também equipamentos tecnológicos altamente avançados, além de prestar serviços de forma inovadora.

As 10 marcas mais valiosas do mundo:

1. Amazon: US\$ 315,5 bilhões (+52%)
2. Apple: US\$ 309,5 bilhões (+3%)
3. Google: US\$ 309 bilhões (+2%)
4. Microsoft: US\$ 251,2 (+25%)
5. Visa: US\$ 177,9 bilhões (+22%)
6. Facebook: US\$ 158,9 bilhões (-2%)
7. Alibaba: US\$ 131,2 bilhões (+16%)
8. Tencent: US\$ 130,8 bilhões (-27%)
9. McDonald's: US\$ 130,3 bilhões (+3%)
10. AT&T: US\$ 108,3 bilhões (+2%)

Fonte: Ranking BrandZ™

Figura 1. Ranking BrandZ™ 2019 das empresas de maior valor de mercado do mundo¹.

Novos modelos de negócio surgiram no mercado, baseados exclusivamente no uso do software, citamos como exemplos o Uber, que é a maior empresa de transporte terrestre de passageiros do mundo e não possui sequer um automóvel próprio para locomoção de passageiros, o AirBnb, maior empresa de acomodação do mundo e que não possui um quarto sequer para acomodação, entre outros.

O uso acentuado do software, inclusive por grandes corporações, está mudando áreas de negócios que são muito tradicionais. Citamos o caso da Ford, uma das empresas mais tradicionais do mundo, que esteve presente ao SXS² (South by Southwest) de 2017, evento de inovação e cultura que acontece nos EUA. Em sua apresentação, Bill Ford, presidente do conselho de administração e bisneto do criador da centenária fabricante de carros, contou ao público do evento o que a empresa aprendeu com as startups e como vê o

¹ Disponível em: <https://economia.uol.com.br/noticias/redacao/2019/06/11/amazon-google-apple-pesquisa-kantar-marcas-mais-valiosas-mundo.htm>. Acesso em: 02/07/2019.

² South by Southwest (SXS[®]) é um conjunto de festivais de cinema, música e tecnologia que acontece toda primavera em Austin, Texas, Estados Unidos. <https://www.sxsw.com>.

futuro das montadoras de carros. Na reportagem que tem como título “A indústria de carros vai se transformar numa indústria de software”, Ford afirma que “as montadoras precisam mudar e virar fábricas de software. Esse é nosso futuro”.³

A nossa dependência pelo uso do software só tende a crescer em ritmo cada vez mais acelerado, e em praticamente todas as áreas do conhecimento humano, sob as mais diversas formas, muitas dessas sequer imaginávamos seria possível um dia. O software está mudando as pessoas, mudando os relacionamentos, mudando os negócios, mudando o mundo, por isso é-a nossa preocupação é em como esse software está sendo produzido/desenvolvido.

1.2. A Crise de Software

Na Conferência da OTAN de 1968, a partir da gravidade dos problemas que estavam sendo discutidos, foi usada pela primeira vez a expressão "crise de software". Os problemas apresentados diziam respeito à baixa qualidade, requisitos não atendidos e estouro de prazo e custo sendo atribuídos à desestruturação de seus desenvolvedores, que não seguiam padrões nem regras de implementação. Os projetos de desenvolvimento de software apresentavam, naquela ocasião, uma grande dificuldade na sua gestão e manutenção (RANDELL, 1995). E isto acontecia em uma época em que a indústria de hardware corria a pleno vapor e demandava sistemas cada vez mais complexos, seguros e eficientes (ABNT; SEBRAE. 2012).

A crise de software referia-se à constatação de que o software demorava mais para se desenvolver do que o estimado, custava mais para se desenvolver do que o orçado e não funcionava muito bem quando finalmente era entregue. No entanto a crise inicial parece ter sido resolvida, embora não tenha havido nenhum avanço definitivo, que tenha proporcionado uma melhoria de ordem de grandeza na produtividade do desenvolvimento de software, uma infinidade de avanços tem sido feita de formas mais incrementais e o software é rotineiramente desenvolvido em grande parte em tempo, dentro do orçamento e funciona bem (FITZGERALD, 2012).

³ Disponível em: <http://revistapegn.globo.com/Startups/noticia/2017/03/industria-de-carros-vai-se-transformar-numa-industria-de-software.html>. Acessado em: 20/10/2017.

A Conferência da OTAN de 1968 também dedicou atenção considerável a muitas questões que nos são bastante familiares hoje em dia. Assim, a questão de como criar processos que poderiam ser eficazes na produção de software de alta qualidade dentro do cronograma e dentro do orçamento foi destacada (MARTINO, 1983). Entre essas muitas questões, se destacam (ABNT; SEBRAE. 2012):

- **Precariedade nas previsões e planejamentos** – os projetos de software atrasam e sofrem problemas de custo por falta de planejamento e controle porque não se prevê adequadamente quanto tempo e esforço serão necessários para produzi-los de maneira que satisfaça as necessidades (requisitos) dos seus clientes.
- **Baixa qualidade de processos e produtos** – a falta de planejamento e de previsibilidade leva a prazos estourados e a produtos de software que por vezes não atendem às necessidades do cliente ou atendem às necessidades que não foram solicitadas originalmente.
- **Requisitos mal definidos** – os requisitos frequentemente não são especificados e, quando o são, ou não estão completos ou apresentam contradições. A garantia de qualidade neste cenário é uma tarefa de “tentativa e sorte”.
- **Alto custo para manutenção** – o que foi produzido não foi bem especificado e tampouco bem documentado; a manutenção corretiva - quando ocorrem erros ou falhas – é difícil de ser identificada. Normalmente isto acontece já em fase de implementação, onde se tem que contabilizar não só o custo do retrabalho, como também o custo de todo o esforço que foi gasto em vão. As manutenções evolutivas, embora sejam novas características adicionadas ao sistema, também podem ter o seu custo onerado quando se trata de um produto de baixa qualidade, carente de especificações e documentações. Não é incomum manutenções tornarem-se inviáveis devido às grandes dificuldades e aos altos custos de implementação.

No entanto, estamos vivendo uma nova crise de software, que FITZGERALD, (2012), denomina de "Crise de Software 2.0". Basicamente surge como resultado da incapacidade de produzir software para alavancar o aumento absolutamente impressionante no volume de dados sendo gerado, por sua vez aliado à enorme quantidade de poder computacional oferecida pelos muitos dispositivos de hardware também disponíveis, e

ambos complementados pelo apetite do recém-surgido consumidor "nativo digital" – que são aqueles que foram imersos no mundo da tecnologia desde o nascimento e desenvolveram uma fluência natural para a tecnologia (ver Figura 2).

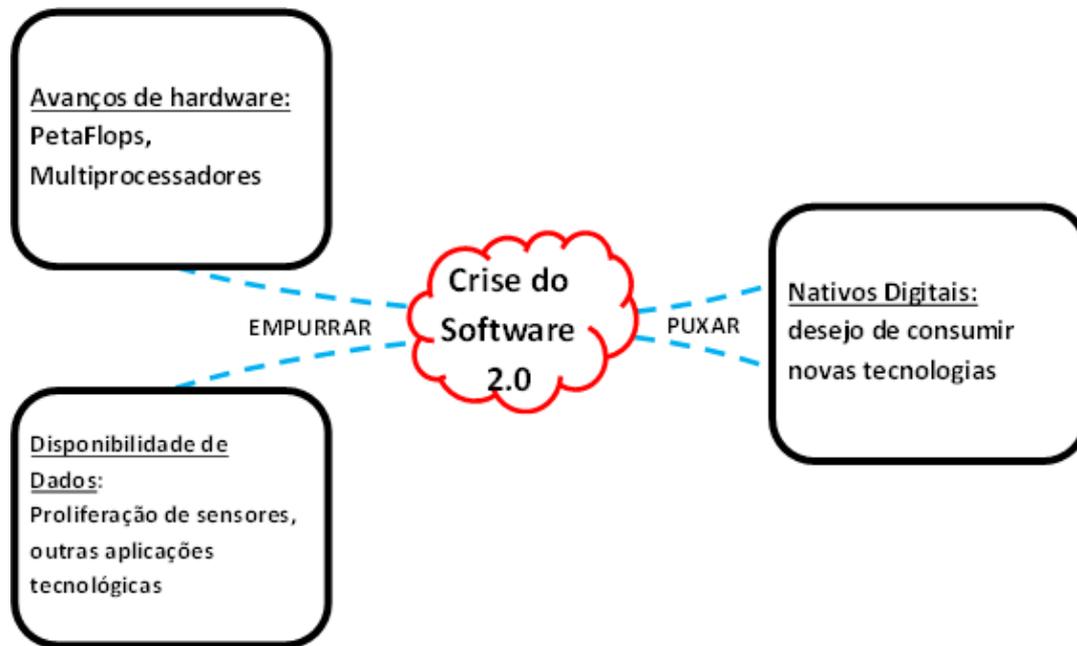


Figura 2. Crise de Software 2.0. Fonte: (FITZGERALD, 2012).

1.3. Problema de Pesquisa

No Primeiro Seminário de Dagstuhl sobre Direções Futuras na Engenharia de Software, realizado em fevereiro de 1992, em Karlsruhe, Alemanha, que teve como principal objetivo “reunir alguns dos principais cientistas na área de software para discutir futuras direções em Engenharia de Software”, que teve como resultado a publicação de uma coletânea dos artigos apresentados pelos cientistas, Phyllis G. Frankl, afirma no artigo “Evolução das Técnicas de Engenharia de Software”, que “Embora muitas técnicas de Engenharia de Software tenham sido propostas, relativamente poucas evidências sólidas foram reunidas para indicar quais delas são efetivas. Novas técnicas são frequentemente justificadas apenas por apelos à intuição.” (TICHY et al, 1992). No mesmo Seminário, David Barstow afirma no artigo “Problemas para engenharia de software”, que “Ainda não existe um conjunto bem caracterizado e amplamente utilizado de ‘práticas padrão de engenharia de software’. É provável que as pressões competitivas no mundo comercial aumentem o uso de boas práticas: os grupos que não usarem boas práticas provavelmente fracassarão em nível financeiro”.

A evolução tecnológica provocou verdadeiras revoluções no processo produtivo de algumas indústrias, como, por exemplo, nas comunicações e na medicina, entretanto, não alcança resultados de tão grande destaque quando se trata de uma das indústrias na qual ela própria é protagonista: a indústria de software (ABNT; SEBRAE, 2012).

A produtividade no desenvolvimento de software tem sido uma importante área de pesquisa há várias décadas. É a chave para uma empresa de software de sucesso controlar e melhorar sua produtividade. No entanto, em contraste com o trabalho industrial tradicional, é difícil medir o desenvolvimento de software (WAGNER, RUH, 2018).

Um software de alta qualidade requer habilidades específicas e a adoção de boas práticas de desenvolvimento e de operação controladas. Nesse sentido a engenharia de software tem a missão de oferecer as ferramentas e métodos adequados para orientar os usuários em todas as atividades ligadas ao ciclo de vida de software, através do uso de tecnologias e novos paradigmas, garantindo produtividade de processos e qualidade de software que envolve performance, disponibilidade, evolução, confiabilidade, etc. (CASALEA et al, 2016). Segundo MADNI et al (2017), um modelo em engenharia de software cria facilidade de implementação de software.

Um princípio fundamental das abordagens baseadas em processos para gerenciamento de engenharia de software é que a qualidade do processo influenciará a qualidade, o custo e o tempo de lançamento do software produzido. Para tais abordagens, surge uma questão crítica sobre o que constitui qualidade para um processo de engenharia de software, tanto em geral quanto dentro do contexto de um determinado ambiente de negócios. As tentativas de responder a essa pergunta são significativamente mais dificultadas pelas visões frequentemente divergentes dos stakeholders do processo de engenharia de software sobre os atributos que constituem a qualidade de um determinado processo. Isso representa um desafio significativo para os especialistas em melhoria de processos encarregados de desenvolver e refinar processos que equilibram as necessidades dessas partes interessadas. O que complica ainda mais a situação é que as visões das partes interessadas raramente são explicitamente declaradas e que os métodos existentes para avaliação do processo de software são, portanto, forçados a adotar visões genéricas ou muito restritas da qualidade do processo (KROEGER et al, 2013).

A Conferência da OTAN de 1968 introduziu e abordou problemas de escala, mas, apesar dos esforços vigorosos ao longo dos últimos 40 anos, muitos dos problemas de 1968 permanecem essencialmente sem solução. Certamente, tem havido um progresso gratificante em áreas como melhoria de linguagem de programação, abordagens de teste de software, suporte para fases de desenvolvimento de pré-codificação e métricas de software. Mas em cada área, as soluções atualmente disponíveis estão longe de ser adequadas para solucionar os problemas enfrentados pelos desenvolvedores de software. (OSTERWEIL, 2007).

O processo de melhoria no desenvolvimento de software passa pela adoção de procedimentos adequados e confiáveis dos métodos, que envolvem os processos, técnicas e ferramentas utilizadas na Engenharia de Software, visando garantir qualidade, integridade e operabilidade com baixo custo de produção e entregas dentro dos prazos estabelecidos. Mas nem sempre é possível para as empresas que produzem software se adaptarem às novas tecnologias e métodos de trabalho, por vários motivos.

Por um lado temos a dependência cada vez maior do uso do software pela sociedade, e por outro, nos deparamos com o problema de desenvolver o software cada vez com mais qualidade pelo custo orçado e dentro de prazos planejados, que foram os principais problemas relatados que levaram a “crise do software” em 1968, e, mais recentemente, outros problemas tem surgido pela crescente evolução tecnológica, que levou a “crise de software 2.0”. Sendo assim, se faz necessária realizar uma pesquisa para investigar esse fenômeno, buscando compreender melhor essa situação e encontrar possíveis soluções para essas questões.

Segundo Gil (2007, p.17 Apud GERHARDT, 2009), pesquisa é definida como o:

(...) procedimento racional e sistemático que tem como objetivo proporcionar respostas aos problemas que são propostos. A pesquisa desenvolve-se por um processo constituído de várias fases, desde a formulação do problema até a apresentação e discussão dos resultados.

Uma alternativa viável, como forma de encontrar soluções, é nos antecipar ao futuro, buscando tendências que indiquem caminhos para melhorar ou aperfeiçoar os métodos de desenvolvimento de softwares. Como métodos entendemos o que

GERHARDT, (2009) afirma, que os métodos dizem respeito aos procedimentos; sendo assim iremos investigar os procedimentos adotados para o desenvolvimento de software.

Para balizar a pesquisa de tendências futuras, poderíamos pensar um futuro para qualquer período de tempo, mas a incerteza futura aumenta à medida que nos afastamos do presente e olhamos para o futuro. Desta forma, imaginamos o futuro para o período de 10 anos. Sendo assim, nosso trabalho se concentrará na seguinte questão:

Que fatores mais influenciarão os métodos de desenvolvimento de software nos próximos 10 anos?

1.4. O Futuro

Para projetar o futuro, parece necessário entender o passado – e o presente também. Conhecer o passado e o presente indicará exatamente onde estamos e também sugerirá o impulso que nos levará para frente (OSTERWEIL, 2007).

A tentativa de antecipar o futuro há muito vem sendo exercitada pelas organizações, no intuito de ganhar tempo em preparar-se para superar seus competidores e, com isso, conquistar espaço na arena competitiva. Uma vez que não é possível prevê-lo de um todo, reflexões acerca de ameaças e oportunidades do porvir são suportadas e direcionadas por técnicas variadas, estudadas cada vez com maior consistência e especificidade de aplicações (SCHENATTO et al, 2011).

Qualquer organização que possa ser afetada por mudanças tecnológicas inevitavelmente se envolve na previsão de tecnologia com cada decisão que aloca recursos para fins particulares, toda a justificativa para produzir previsões será seu uso na tomada de decisões. A implicação disso é que usar as previsões ajuda a tomar melhores decisões. Em particular, as previsões desempenham papéis específicos na melhoria da qualidade da tomada de decisões (MARTINO, 1983).

Para tomar decisões estratégicas às empresas precisam estar preparadas para tal, baseando-se no resultado de tendências, em informações que possam possibilitar a escolha adequada entre alternativas viáveis. A qualidade da decisão não pode ser medida pelo resultado, mas tão somente pelo modo como se chegou a ela, isto é, por quão habilidoso e vigilante se foi ao se tomar a decisão (HEIJDEN, 2009).

Janis (Janis & Mann 1977, apud HEIJDEN, 2009) sugere que uma decisão é vigilante se:

- As razões são racionais (ou seja, são explícitas e compreensíveis, têm coerência lógica e são congruentes com o conhecimento existente);
- As razões baseiam-se em uma busca adequada por dados empíricos relevantes;
- As razões levam em conta a indeterminação futura (planejamento de contingência).

A decisão pela adoção da **prospecção** ou do **estudo prospectivo** deve levar em conta a comparação entre as suas características em relação às características da **previsão** ou do **estudo tendencial**, conforme podemos observar no Quadro 1.

Característica	Previsão ou estudo tendencial	Prospecção ou estudo prospectivo
Compreensão do fenômeno	Em parte “todas as demais condições iguais”	No todo “nada permanecendo igual”
Variáveis	Quantitativa, objetiva e conhecida	Qualitativa, subjetiva, conhecida ou não
Relações	Estáticas, estruturas fixas	Dinâmicas, estruturas em evolução
Explicação	O passado explica o futuro	O futuro é a razão de ser do presente
Futuro	Simple e certo	Múltiplo e incerto
Método	Modelos determinísticos e quantitativos (econométricos e matemáticos)	Análise intencional, modelos qualitativos (análise estrutural) e estocásticos.
Atitude em relação ao futuro	Passiva e adaptativa (o futuro emerge)	Ativa e criativa (o futuro é construído)

Quadro 1. Comparativo das características entre a Previsão e a Prospecção (SCHENATTO et al, 2011).

A prospecção pode ser entendida como nossa capacidade de usar e criar conhecimento de futuros, bem como de imaginar futuros possíveis; ver futuros alternativos como recipientes candidatos para gerenciar conteúdo histórico sob as restrições do presente é a mentalidade de previsão, retrospectiva e insight (AALTONEN, BARTH, 2005).

O nosso estudo tem como características principais que as variáveis estudadas são de natureza qualitativa, pois não existem dados passados precisos que possam ser usados para fazer previsões para o futuro, são subjetivas e desconhecidas, em razão do desenvolvimento tecnológico em constante evolução afetando constantemente a forma como o software é desenvolvido; as relações são dinâmicas; o futuro para a engenharia de software, assim como para o próprio software é múltiplo e incerto; e a atitude em relação ao futuro é ativa e altamente criativa, estando em constante construção pelo surgimento cada vez mais crescente de novas tecnologias.

A prospecção tecnológica se torna um fundamento essencial para que as empresas de software possam melhorar seus métodos de desenvolvimento de software, melhorando a produtividade, e possam se preparar para enfrentar a concorrência cada vez mais intensa, se antecipando na tomada de decisões, e se tornando capazes de lidar com as novas tecnologias da melhor forma e em todo o seu potencial, além de se prepararem para a adoção de novos métodos de engenharia de software, que envolvem também novos processos, técnicas e ferramentas, de forma tranquila e no tempo necessário para tal.

1.5. Objetivos

O objetivo geral deste trabalho é construir, a partir de tendências, cenários futuros que impactarão os métodos de desenvolvimento de software para os próximos dez anos.

Os objetivos específicos são:

1. Investigar na literatura problemas persistentes nos métodos de desenvolvimento de software relacionados à engenharia de software;
2. Aplicar um método de Estudo de Futuro para identificar tendências que influenciarão os métodos de engenharia de software nos próximos 10 anos;
3. Avaliar os resultados e preparar cenários que possam ser úteis do ponto de vista da tomada de decisão no processo de desenvolvimento de software no futuro.

1.6. Organização da Dissertação

Os próximos capítulos da dissertação estão organizados da seguinte forma:

Capítulo 2 – Referencial Teórico. A finalidade deste capítulo é ter conhecimento sobre estudos do futuro, entender o que é o software e quais são as suas

principais características, além de entender os métodos, bem como as práticas, processos e metodologias de engenharia de software atuais.

Capítulo 3 – Metodologia da Pesquisa. Este capítulo apresenta as etapas da pesquisa, a escolha da metodologia, bem como as fases de aplicação da metodologia adaptada do modelo de WOHLIN & AURUM (2014).

Capítulo 4 – Resultados da Pesquisa. São apresentados o resultado da aplicação dos questionários do método Delphi e a construção de Cenários possíveis para o futuro dos processos de desenvolvimento de software.

Capítulo 5 – Conclusões e Trabalhos futuros. Este capítulo apresenta os principais resultados e conclusões referentes à pesquisa realizada nesta dissertação, bem como as contribuições nos principais aspectos que influenciarão a engenharia de software no futuro. Por fim, são apresentadas algumas perspectivas em aberto para trabalhos futuros.

2. Referencial Teórico

Este capítulo apresenta a base conceitual necessária para compreensão e desenvolvimento do trabalho, relacionando as principais áreas do estudo: Estudos do Futuro, Conceitos sobre o Software, as principais características e conceitos da Engenharia de Software, e os principais Métodos de Desenvolvimento de Software atuais.

2.1. Estudos do Futuro

Estudos do futuro são um mosaico de abordagens, objetivos e métodos, e muitas partes dele estão em diferentes estágios de evolução (KUOSA, 2009).

Uma abordagem possível para categorizar orientações de pesquisa de futuros é o método que Borg usa. Ele não divide a pesquisa futura em paradigmas abrangentes. Em vez disso, ele se concentra em grandes áreas de pesquisa de futuros que têm objetivos de pesquisa diferentes. Borg afirma que se a orientação da predição antiga e a imaginação moderna da utopia/distopia são consideradas como uma abordagem unificada, ela pode ser descrita como a **primeira** grande área de objetivos de pesquisa de futuros. Essa seria a criação de interessantes imagens futuras, visões e cenários. A **segunda** grande área de objetivos de pesquisa de futuros é a sua capacidade de apoiar o planejamento e a tomada de decisões. Aqui, sua aplicabilidade no planejamento é o ponto focal. A **terceira** grande área de objetivos de pesquisa de futuros está resolvendo as grandes questões globais de toda a humanidade. De acordo com Borg, Flechtheim foi o pioneiro na definição das questões e objetivos desta terceira grande área problemática. Finalmente, Borg define uma **quarta** grande área de objetivos de pesquisa de futuros como o desenvolvimento de metodologia interdisciplinar aplicável (KUOSA, 2009).

Busca-se, a partir do referencial tecnológico, estudar as características das trajetórias tecnológicas consolidadas e identificar possíveis desdobramentos e principais condicionantes, além de identificar trajetórias emergentes e/ou alternativas. Nesse caso, por meio da gestão da informação se pode visualizar o estado da arte e as tendências de determinado setor ou tema, com o objetivo de gerar informações sobre a sua trajetória passada e sobre as perspectivas futuras, bem como emitir a percepção sobre tendências inovadoras não consensuais (SANTOS et al, 2004).

Segundo KUOSA (2009), se nos concentramos em futuristas que usam a palavra paradigma e que afirmam que haverá algum tipo de mudança de paradigma na pesquisa de futuros, uma das apresentações mais sólidas é feita por Mannermaa. Ele tenta dividir o campo de pesquisa em três paradigmas simultâneos e alternativos. Seus paradigmas são os seguintes:

- **O paradigma descritivo.** Isso se refere a uma tentativa de apresentar previsões altamente prováveis que se baseiam no desenvolvimento observado no passado. Aqui, a visão para o futuro é estática e otimista. Acredita-se que o futuro seja algo que possa ser previsto. O objetivo da pesquisa não é turbulento, os métodos são principalmente quantitativos e o tempo é curto.
- **O paradigma do cenário.** Isso se refere a uma tentativa de descrever os diferentes scripts manuais para o futuro. O valor dos cenários não se baseia na sua capacidade de prever qualquer coisa, exceto a sua capacidade de auxiliar a tomada de decisão atual, imaginando o que é possível e fazendo descobertas interessantes sobre o possível desenvolvimento.
- **O paradigma evolutivo da pesquisa de futuros.** Isso se refere a uma tentativa de descrever e entender os futuros no mundo turbulento de forma mais precisa e basear-se em leis evolutivas. Baseia-se principalmente nas descobertas da pesquisa de complexidade e no reconhecimento da evolução em geral.

2.1.1. Método Delphi

O método Delphi, cujo nome é uma referência ao oráculo da cidade de Delfos na Antiga Grécia, começou a ser idealizado em 1948 por Dalkey, Gordon, Helmer e Kaplan que produziram 14 documentos que são considerados o preâmbulo do método. Utiliza as diversas informações identificadas e obtidas pelo julgamento intuitivo das pessoas, com a finalidade de delinear e realizar previsões. (Oliveira, 2001 Apud SANTOS et al 2004).

Foi desenvolvida durante a década de 1950 pelos trabalhadores da RAND Corporation, enquanto participavam de um projeto patrocinado pela Força Aérea dos EUA. O objetivo do projeto era a aplicação da opinião de especialistas à seleção - do ponto de vista de um planejador estratégico soviético - de um sistema ideal de destino industrial dos EUA, com uma estimativa correspondente do número de bombas atômicas necessárias para

reduzir a produção de munição, uma quantidade prescrita. Mais genericamente, a técnica é vista como um procedimento para "obter o mais confiável consenso de opinião de um grupo de especialistas" por uma série de questionários intensivos intercalados com feedback de opinião controlada "(Dalkey & Helmer, 1963, p. 458 apud ROWEA & WRIGHTB, 1999).

O Método Delphi foi projetado para fornecer os benefícios de uma “votação” e a troca de opiniões, para que os entrevistados possam aprender com os pontos de vista de cada um, sem o tipo de influência indevida provável em configurações face a face convencionais (que são tipicamente dominadas pelas pessoas que falam mais alto ou têm mais prestígio). A técnica permite que especialistas lidem sistematicamente com um problema complexo. De rodada para rodada, as informações relevantes são compartilhadas, instruindo ainda mais os membros do painel. As recomendações podem, assim, ser feitas com base em informações mais completas (SLOCUM, 2005).

O método de opinião de especialistas tem seus limites estabelecidos naquilo que as pessoas percebem como factível, de acordo com sua imaginação e crenças, e deve ser usada sempre que a informação não puder ser quantificada ou quando os dados históricos não estão disponíveis ou não são aplicáveis. Mesmo quando há dados históricos, a opinião de especialistas pode e deve ser usada como uma forma de complementar as informações obtidas e de captação de conhecimentos tácitos, sinais fracos e insights. Por isso, tais métodos são considerados qualitativos.

A pesquisa Delphi geralmente é usada para a coleta de opiniões subjetivas porque reúne as sugestões que os especialistas têm em questões específicas. Em seguida, analisa essas opiniões para examinar se os peritos alcançam um consenso sobre uma questão específica. Caso contrário, temos outra rodada de pesquisas até que não haja muita voz para as questões específicas (CHENG et al., 2008 Apud LEE et al 2012).

Segundo MARTINO (1983), os métodos de previsão formais destinam-se a substituir a opinião subjetiva com dados objetivos e métodos replicáveis. No entanto, existem três condições nas quais a opinião de especialistas será sempre necessária.

- A primeira condição é quando não existem dados históricos. No planejamento tecnológico, isso geralmente envolve novas tecnologias. Apesar da falta de

dados históricos, uma previsão pode ser necessária. O parecer do perito é a única fonte possível de uma previsão.

- A segunda condição é quando o impacto de fatores externos é mais importante do que os fatores que governaram o desenvolvimento prévio da tecnologia. Esses fatores externos podem incluir decisões de patrocinadores e opositores da tecnologia e mudanças na opinião pública. Nesse caso, os dados sobre o passado podem ser irrelevantes. A opinião dos especialistas sobre os efeitos desses fatores externos pode ser a única fonte possível de uma previsão. Os exemplos históricos recentes de tecnologias em que os fatores externos eram dominantes são a tecnologia espacial (decisão do presidente Kennedy para fazer a lua um objetivo) e a energia nuclear (opor por motivos de segurança).
- A terceira condição é quando as considerações éticas ou morais podem dominar as considerações econômicas e técnicas que costumam governar o desenvolvimento da tecnologia. Essas questões são inerentemente subjetivas e a opinião de especialistas pode ser a única fonte possível de uma previsão. Um exemplo, atual a partir desta escrita, é pesquisa sobre tecido fetal obtido por abortos. A tecnologia é direta. O uso dessa tecnologia depende do resultado de um debate sobre a ética desse uso. Prever o futuro da tecnologia só pode ser feito usando a opinião de especialistas sobre o resultado do debate.

O uso de especialistas, segundo a maioria das publicações sobre o Delphi, continua sendo identificado como um critério importante e que confere credibilidade ao método. A principal justificativa para o uso de especialistas reside na crença de que estes sejam formadores de opinião (KAYO & SECURATO, 1997).

Segundo ROWEA & WRIGHTB (1999) as seguintes propriedades, entre outras, levam à necessidade ou utilidade de empregar o Delphi:

- O problema não se presta a técnicas analíticas precisas, mas pode se beneficiar de julgamentos subjetivos em uma base coletiva.
- Os indivíduos que precisam contribuir para o exame de um problema amplo ou complexo não têm histórico de comunicação adequada e podem representar origens diversas no que diz respeito à experiência ou especialização.
- São necessários mais indivíduos do que podem interagir efetivamente em uma troca face a face.

- A heterogeneidade dos participantes deve ser preservada para assegurar a validade dos resultados, ou seja, evitar a dominação pela quantidade ou pela força da personalidade.

Esse método explora a experiência coletiva dos membros de um grupo em um processo interativo e estruturado. No formato original, a primeira rodada é não estruturada, e é dada aos especialistas selecionados uma relativa liberdade de identificar e elaborar as questões percebidas como relevantes ao tema abordado. O questionário é consolidado pela equipe de coordenação, de modo a associar escalas qualitativas ou quantitativas às questões, e então submetê-lo a uma sequência de rodadas, conforme demonstrado pelo fluxograma da Figura 3.

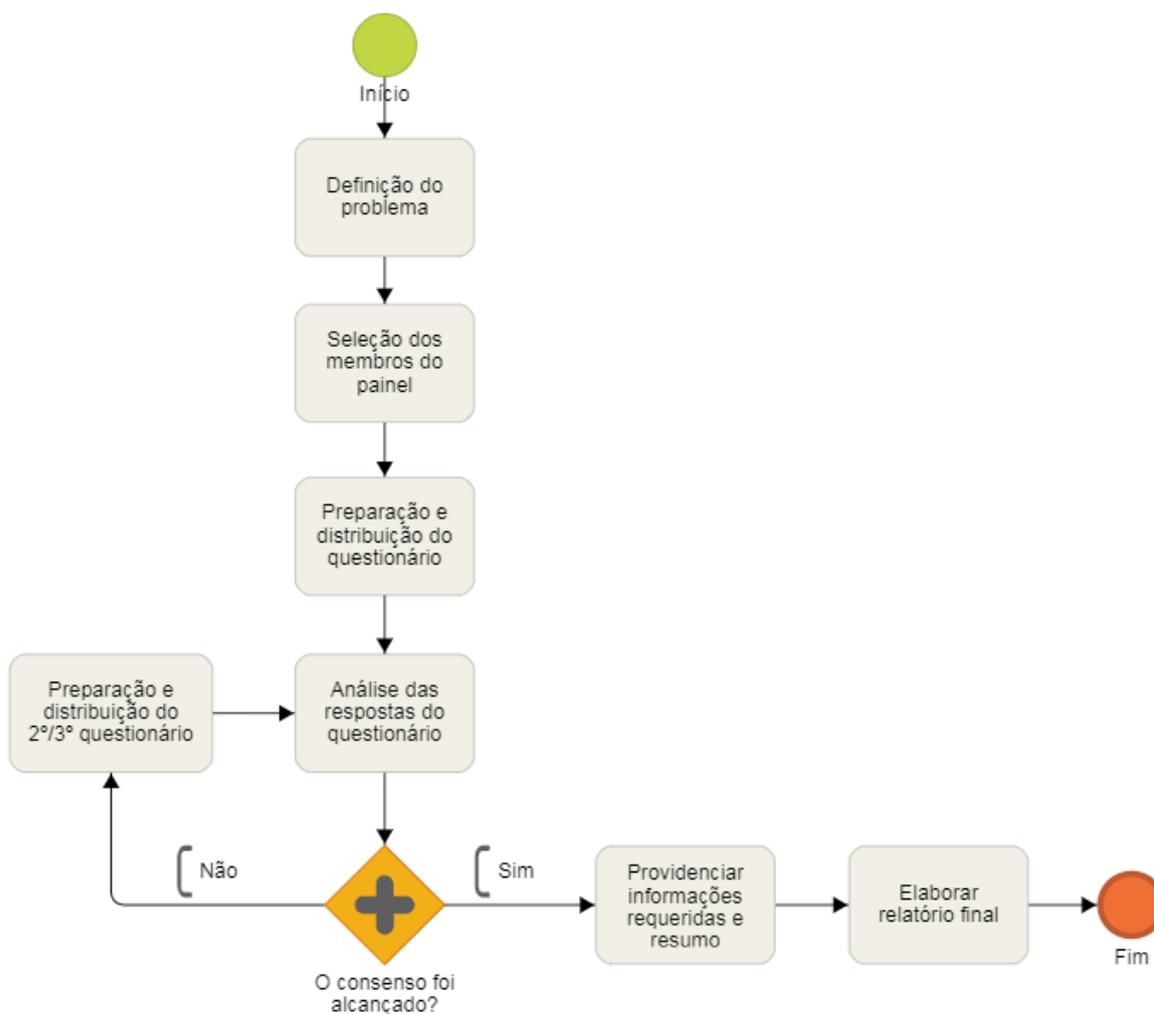


Figura 3. Fluxograma de aplicação do método Delphi.

O Delphi – convencional, online ou pessoal – geralmente passa por quatro fases. Na *primeira* fase, o assunto em discussão é explorado e cada indivíduo contribui com informação (ou informações) que considera pertinente para a questão. Na *segunda* fase, uma visão geral é alcançada sobre como o grupo vê a questão, por exemplo, onde há discordância sobre o significado de termos relativos como "factível", "importante", "desejável" etc. É um desacordo significativo, então isso é explorado na *terceira* fase, a fim de esclarecer as razões para as diferenças e avaliá-las. A *quarta* fase envolve uma avaliação final que ocorre quando todas as informações previamente coletadas foram inicialmente analisadas e as avaliações foram realimentadas para reconsideração (ROWEA & WRIGHTB, 1999).

2.1.1.1. Delphi Online

O Delphi Online ou WebDelphi é uma ferramenta para a prospecção de futuro e formulação de estratégias, em grupo, por meio da internet (SANTOS, 2004). Tem como vantagens em relação ao Delphi tradicional: a redução de custos, pelo uso da internet ao invés do preenchimento de questionários e divulgação da consolidação das opiniões; o alcance geográfico, pois especialistas de quaisquer partes do mundo podem participar ativamente e simultaneamente do processo; e a redução do tempo da pesquisa, pois elimina o atraso causado na sumarização de cada rodada do Delphi, transformando o processo em um sistema de comunicação em tempo real.

Baseia-se no método Delphi tradicional, de previsão por meio de consultas a especialistas. É indicado para situações de mudanças estruturais, inexistência de dados históricos ou horizontes de tempo muito longos. A pesquisa é interativa, caracterizada pelo feedback e convergência a uma visão representativa dos especialistas consultados (SANTOS, 2004).

A principal diferença entre o Delphi tradicional e o Delphi online é o uso de uma ferramenta pela internet para aplicação do questionário. No entanto, as atividades de formulação do questionário, da escolha dos especialistas, das rodadas de aplicação e da análise dos dados, se mantêm idênticos em ambos os métodos.

Segundo GIOVINAZZO & FISCHMANN (2001) existe uma sequência básica de atividades envolvidas na aplicação do Delphi online (eletrônico):

- O objetivo de pesquisa deve ser claramente definido, especificando o horizonte de tempo e o tipo de resultado desejado. A equipe coordenadora do Delphi deve procurar informações sobre o tema, recorrendo à literatura especializada e a entrevistas com técnicos do setor. Deve então estruturar um primeiro modelo do questionário;
- As questões devem ser cuidadosamente elaboradas. Em função das necessidades específicas do estudo, diferentes tipos de questões podem ser utilizados. Nesta fase é importante a interação entre os coordenadores do estudo e especialistas do setor, para assegurar a correção técnica das questões formuladas;
- Enquanto o questionário da primeira rodada é desenvolvido e testado via internet, a equipe coordenadora faz a seleção dos painelistas. Em geral deve-se buscar uma distribuição equilibrada entre elementos de dentro e de fora da entidade interessada, recorrendo-se a universidades, institutos de pesquisa, indústrias e outros setores da sociedade. A qualidade do resultado depende essencialmente dos participantes do estudo. Um número de 15 a 30 painelistas é considerado um bom número, o suficiente para gerar informações relevantes, embora grupos maiores possam ser utilizados com sucesso;
- O questionário é disponibilizado no site, utilizando um formulário cujos dados das respostas são encaminhados para uma planilha eletrônica como Excel, ou outra planilha adequada para a tabulação das informações. Os respondentes são comunicados por e-mail e preenchem o questionário diretamente pela internet;
- Após um prazo determinado pela coordenação, a equipe procede à sua tabulação e análise, calculando a mediana e os quartis, e procurando associar os principais argumentos às diferentes tendências das respostas. Após a análise da 1ª rodada a coordenação deve decidir sobre a necessidade de incorporação de novas questões na 2ª rodada, o que é bastante comum;
- A segunda rodada do questionário Delphi apresenta obrigatoriamente os resultados do primeiro questionário, possibilitando que cada respondente reveja sua posição face à previsão e argumentação do grupo, em cada pergunta. Por incluir os resultados da rodada anterior e, ocasionalmente, novas questões, o segundo questionário geralmente é mais extenso que o primeiro. As questões em geral

objetivam a convergência de resultados da primeira rodada, e são rediscutidas à luz da argumentação dos painelistas. Novos temas são explorados ou sugeridos e discutem-se possíveis incompatibilidades entre as tendências previstas. A segunda rodada é, da mesma forma que a primeira, disponibilizada na internet. As rodadas sucedem-se até que seja atingido um grau satisfatório de convergência. No mínimo, duas rodadas são necessárias para caracterizar o processo Delphi, sendo raros os exemplos de estudos com mais de três rodadas de questionários;

- Os resultados finais são processados a partir do consenso gerado pelo grupo. Estes resultados, sintéticos ou completos, interpretação de dados e considerações levantadas no processo podem ser divulgados na internet para acesso público ou restrito aos participantes, mediante utilização de uma senha, conforme os objetivos da pesquisa.

2.1.2. Método de Construção de Cenários

KAHN (1967), considerado um dos fundadores dos estudos de futuros e pai do planejamento de cenários, define cenário como sendo “*um conjunto de eventos hipotéticos estabelecidos no futuro construídos para esclarecer uma possível cadeia de eventos causais, bem como seus pontos de decisão*”.

O **cenário** não prevê o futuro, mas explora múltiplas situações futuras plausíveis com o objetivo de ampliar a esfera de pensamento dos participantes no processo de desenvolvimento de cenários. Os cenários são diferentes das previsões, pois, nos cenários, uma série de possíveis resultados resultantes da incerteza são explorados, enquanto que nas previsões o objetivo é identificar a via mais provável e estimar as incertezas (AMER et al, 2013).

Segundo HEIJDEN (2009), em projetos práticos de cenários, entre várias possibilidades, alguns são feitos para convergir e focar em torno de uma estratégia, e alguns visam desenvolver habilidade de antecipação. Alguns autores, de acordo com Mietzner e Reger (2004 Apud HEIJDEN, 2009), fazem distinção entre construção e planejamento de cenários. A construção de cenários leva em consideração as incertezas que cercam o futuro: avaliam e identificam possíveis resultados para futuros diferentes. Nesta concepção, a construção de cenários é a fundação necessária para o Planejamento de Cenários, uma metodologia de administração usada por gerentes para articular seus

modelos mentais sobre o futuro e, desta forma, melhorar a tomada de decisão. Outros não distinguem cenários do próprio planejamento, eles os unem (JUNIOR et al, 2010).

A redação de cenários propõe diferentes concepções de tecnologia futura. Cada concepção das características da tecnologia futura é baseada em um conjunto bem definido de suposições. Um cenário representa características alternativas da tecnologia futura, sendo cada alternativa baseada em certas suposições e condições. O previsor avalia a validade das suposições. Os resultados dessa avaliação são usados para determinar o cenário mais provável de ocorrer (FIRAT et al, 2008).

Em um mundo que muda muito rapidamente para que a previsão seja precisa, os cenários estão ganhando credibilidade como ferramentas eficazes para a preparação para um futuro incerto, para alterar modelos, para testar decisões e para melhorar o desempenho em um ambiente dinâmico (CHERMACK, 2003).

As abordagens de desenvolvimento de cenários apresentadas por Dator (JUNIOR et al, 2010), de futuros alternativos articula quatro arquétipos de cenários. Tomando estes quatro cenários, pode-se imaginar como seria o futuro em cada um desses cenários. Dator propõe os seguintes arquétipos:

- **Crescimento contínuo:** Neste futuro, supõe-se que as condições e tendências atuais sejam aprimoradas.
- **Colapso:** Este futuro resulta quando o crescimento contínuo falha e existem grandes contradições.
- **Estado estável:** Este futuro procura deter o crescimento e encontrar um equilíbrio na economia e com a natureza. Ele destaca uma sociedade equilibrada, mais suave e mais justa.
- **Transformação:** Este futuro tenta mudar os pressupostos básicos dos outros três. Ele sai através de mudanças tecnológicas dramáticas ou mudanças espirituais.

Vários futuristas e pesquisadores recomendaram um número diferente de cenários alternativos, geralmente variando de três a seis cenários. É fundamental desenvolver um número gerenciável de cenários, de uma maneira lógica, que melhor capte a dinâmica da situação e comunique efetivamente os problemas centrais. Durance e Godet recomendam desenvolver cenários em torno de quatro ou seis hipóteses fundamentais,

porque, caso contrário, a enorme magnitude das combinações possíveis será esmagadora (JUNIOR et al, 2010).

Segundo JUNIOR et al (2010), duas escolas de estudos sobre cenários se destacam no uso e em estudos organizacionais: a Escola da Shell, popularizada por Wack, Schwartz e Van der Heijden; e a Prospectiva que tem como expoente Michel Godet.

ETAPA	ESCOLA SHELL	ESCOLA DE MICHEL GODET
A	Definição do Problema, questão ou decisão a ser tomada.	Definição do Problema, questão ou decisão a ser tomada.
B	Fatores-chave	Árvore de competências (diagnóstico)
C	Forças motrizes do macroambiente	Variáveis-chave
D	Hierarquização das variáveis; incerteza e importância.	MACTOR
E	Definição de eixos ortogonais	Consulta a especialistas: Delphi
F	Construção de cenários (narrativas)	Opções estratégicas
G	Opções estratégicas	Opções x Incerteza
H	Indicadores para monitoramento	Estratégias e Objetivos
I	-	Planos de ação e monitoramento

Quadro 2. Convergências e divergências dos Modelos de Planejamento de Cenários da Shell e da Escola Francesa (JUNIOR et al, 2010).

As duas escolas apresentam a seguinte sequência de desenvolvimento:

- **Primeira etapa:** em ambos os modelos é a mesma, ou seja, parte-se de um problema a ser resolvido, uma questão a ser analisada ou uma decisão a ser tomada.
- **Segunda etapa:** a escola Shell busca levantar os fatores-chave que interferem de maneira positiva ou negativa na questão central, ou decisão a ser tomada. A escola francesa, por sua vez, busca diagnosticar a totalidade da organização na forma de uma árvore: nas raízes se encontram as competências técnicas e Know How; no tronco a capacidade de produção; nos ramos as linhas de produto e mercados. Para Godet, a árvore de competências é essencial para a determinação das forças e fraquezas de uma organização, nas palavras do autor, “para saber aonde quer ir é preciso saber de onde se vem”.
- **Terceira etapa:** a escola francesa busca selecionar as variáveis chaves do ambiente do sistema em estudo tanto interno quanto externo. Para Godet, esta

lista de variáveis não deve exceder 70-80 observações. Após o levantamento das variáveis, as mesmas são submetidas - aos pares - à avaliação da influência de uma sobre a outra, pontuando-se esta influência em: (0) não existe, (1) fraca influência, (2) média, (3) forte e (4) potencial. Depois desta etapa, a identificação propriamente dita das variáveis-chave ocorre por meio da classificação direta e indireta. Na escola Shell, esta etapa se atém à identificação de forças motrizes do macroambiente (atividade inexistente na escola francesa).

- **Quarta etapa:** a escola Shell hierarquiza por importância e incerteza as variáveis identificadas. A escola francesa busca, por meio da Metodologia dos Atores, Objetivos e Relações de Força (MACTOR), valorizar as relações de força entre os atores e estudar suas convergências e divergências em relação a determinado número de posturas e objetivos associados.
- **Quinta etapa:** no modelo de Godet, faz-se uso da Metodologia Delphi, que se constitui na consulta a especialistas no tema em questão por meio de questionários sucessivos de maneira a definir com precisão o campo da investigação. Neste ponto, ocorre a construção de cenários, segundo a metodologia da escola francesa. Já a escola Shell define, nesta etapa, os eixos ortogonais que dão origem aos cenários.
- **Sexta etapa:** no modelo da escola Shell, os cenários são construídos utilizando uma relação causal entre fatores e tendências na forma de narrativa, enquanto no modelo de Godet são formuladas as opções estratégicas.
- **Sétima etapa:** o modelo Shell retorna à questão central, definida na primeira etapa, e formula as estratégias. A escola francesa avalia as opções estratégicas em função das incertezas.
- **Oitava etapa:** a escola francesa formula as estratégias e respectivos objetivos. A escola Shell identifica indicadores de cenários a serem monitorados.

Para melhor adequação aos objetivos a serem alcançados por este trabalho, optamos por desenvolver um modelo híbrido, com etapas dos dois modelos, o da Shell e o de Godet (Escola Francesa), com seis etapas, conforme apresentado no Quadro 3.

ETAPA	Atividades	Escola
A	Definição do Problema.	Ambas

B	Fatores-chave	Shell
C	Variáveis-chave	Godet
D	Hierarquização das variáveis; incerteza e importância.	Shell
E	Consulta a especialistas: Delphi	Godet
F	Opções estratégicas	Godet

Quadro 3. Modelo híbrido de Planejamento de Cenários, baseado nos modelos da Shell e da Escola Francesa (Adaptado de JUNIOR et al 2010).

2.2. Software

Apesar de haver muitas definições sobre o que é o software, não há, contudo, uma definição que seja aceita como padrão pela comunidade (OSTERWEIL, 2007), e, para defini-lo, alguns autores precisaram antes explicar o seu uso, ou o seu contexto. Seguem duas definições encontradas na literatura conceituando o software, que consideramos claras e elucidativas:

- O software começou como sinônimo de programas de computador, mas o termo assumiu um significado muito mais extenso. O uso efetivo de hardware de computador requer mais do que programas. Como resultado do papel difundido do software em um sistema de processamento multi distribuído, parece apropriado redefinir o termo software do seu significado usual de programas únicos para significar a doutrina lógica para a cooperação harmoniosa de um sistema de pessoas e máquinas - geralmente muitos tipos de pessoas e muitos tipos de máquinas (MILLS, 1999).
- O software de computador é o produto que os profissionais de software criam e depois suportam a longo prazo. Ele engloba programas executados em um computador de qualquer tamanho e arquitetura, conteúdo apresentado à medida que os programas de computador são executados e informações descritivas em formulários impressos e virtuais que abrangem praticamente qualquer mídia eletrônica (PRESSMAN, MAXIM, 2015).

2.2.1. Ciclo de Vida do Software

Modelado após o ciclo de engenharia convencional, o paradigma do ciclo de vida engloba diversas atividades. A Figura 4 destaca as várias fases do que é provavelmente o mais antigo processo de desenvolvimento de software existente, a saber, o paradigma clássico do ciclo de vida, às vezes chamado de "modelo em cascata". Este paradigma implica uma abordagem sistemática e sequencial (raramente alcançada na prática) para o desenvolvimento de software que começa no nível do sistema e avança através da análise, projeto, codificação, teste e manutenção (WANG, KING, 2000).

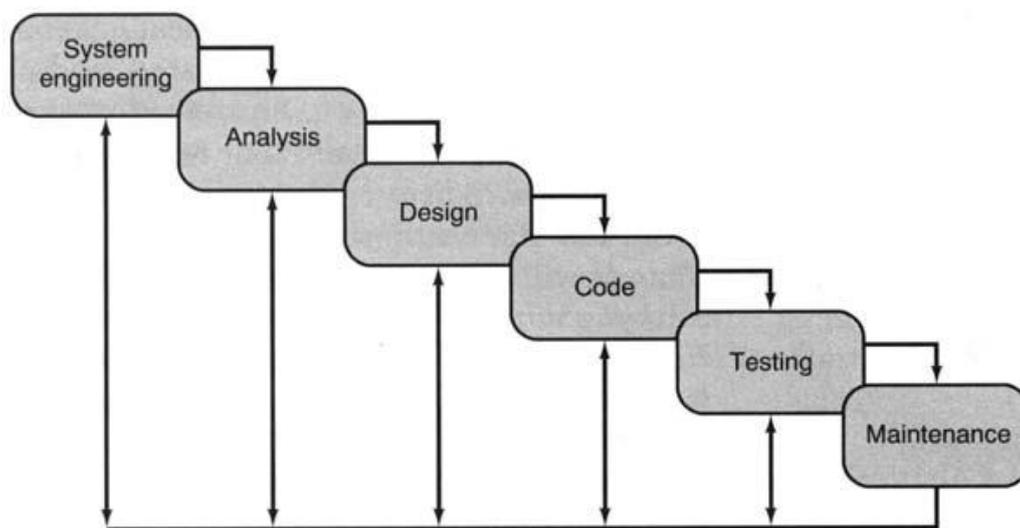


Figura 4. Paradigma clássico do ciclo de vida do software. Fonte: (WANG, KING, 2000).

O ciclo de vida do software é caracterizado por um conjunto de atividades definidas para descrever o processo de desenvolvimento do software, desde a definição do sistema até o suporte operacional. As atividades que compõem o processo de software são definidas em termos de componentes de trabalho que identificam as tarefas a serem executadas, conforme mostrado na Tabela 1. Essas atividades retratam o software na perspectiva ampliada de todo o ciclo de vida, conforme visto pelo cliente, e fornece a base para uma gestão eficaz (MILLS, 1999).

Atividades	Componentes de trabalho
Definição do sistema	Definição de requisitos de software Descrição do sistema de software Planejamento de desenvolvimento de software

Design de software	Design funcional Projeto de programa Projeto de teste Ferramentas de software Avaliação de design
Desenvolvimento de software	Desenvolvimento de Módulos Teste de desenvolvimento
Teste do sistema de software	Procedimentos de teste do sistema de software Integração de software e teste
Sistema e teste de aceitação	Suporte de teste do sistema Suporte para teste de aceitação
Suporte operacional	Suporte de operação do sistema Treinamento Suporte de implantação de site

Tabela 1. Ciclo de vida do software e componentes (MILLS, 1999)

2.2.2. Qualidade do Software

KROEGER et al (2013) conduziram pesquisa com o argumento de que uma visão multifacetada de qualidade é necessária para permitir o desenvolvimento de processos de engenharia de software para equilibrar as necessidades de diversos grupos de partes interessadas. Como resultado da pesquisa, descobriram que os profissionais percebem a qualidade dos processos de engenharia de software em termos de quatro atributos distintos de qualidade: adequação, usabilidade, capacidade de gerenciamento e evolução. Esses atributos de nível superior representam, cada um, uma perspectiva diferente da qualidade e consistem em 15 sub atributos de nível inferior, conforme mostrado na Figura 5.

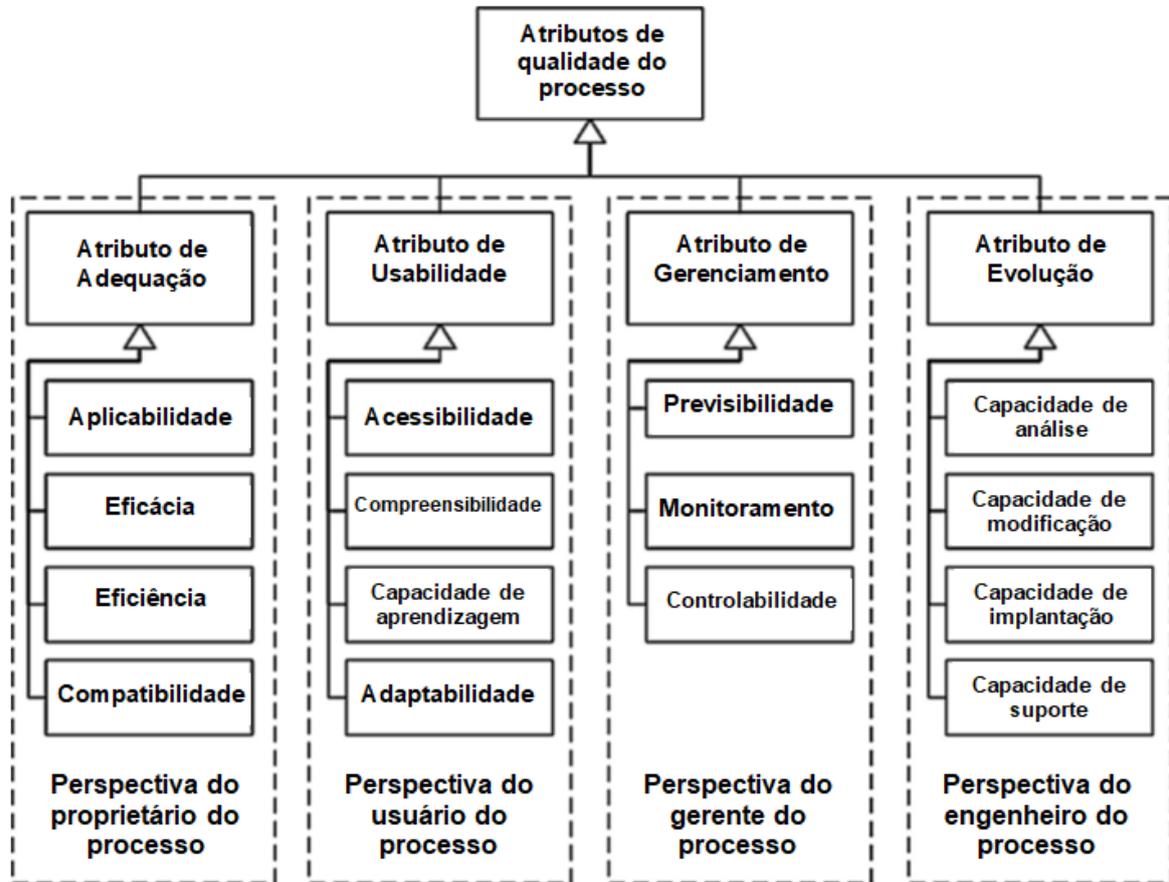


Figura 5. Uma tipologia para atributos de qualidade do processo de engenharia de software (KROEGER et al, 2013).

Por outro lado, o grupo de Garantia de Qualidade de Software (SQA), que é uma organização profissional fundada para promover a profissão de garantia de qualidade através da proliferação e avanço de altos padrões profissionais, adotou uma estrutura como uma diretriz para avaliar a qualidade de um sistema de software. A **Tabela 2** mostra uma versão do modelo de qualidade de software impresso no manual do SQA. Esse modelo identifica 14 fatores de qualidade em três estágios do ciclo de vida de desenvolvimento: design, desempenho e adaptação. Embora seja difícil, se não impossível, desenvolver um software que atenda a todos esses 14 fatores, o modelo fornece um bom quadro de referência para entender a qualidade do software (ASHRAFI, 2002).

Qualidade do design

Correção

Até que ponto o software está em conformidade com suas especificações e está em conformidade com seus objetivos

declarados

Manutenção Facilidade de esforço para localizar e corrigir uma falha de software dentro de um período de tempo especificado

Verificabilidade Facilidade de esforço para verificar recursos de software e desempenho com base em seus objetivos declarados

Qualidade de desempenho

Eficiência Até que ponto o software é capaz de fazer mais com menos recursos de sistema (hardware, sistema operacional, comunicações etc.)

Integridade Até que ponto o software é capaz de suportar a invasão por usuários não autorizados ou software dentro de um período de tempo especificado.

Confiabilidade Até que ponto o software irá executar (de acordo com seus objetivos declarados) dentro de um período de tempo especificado.

Usabilidade Facilidade relativa de aprendizagem e operação do software.

Testabilidade Facilidade de teste para programar para verificar se ele executa uma função especificada.

Qualidade de Adaptação

Expansibilidade Esforço relativo necessário para expandir os recursos e / ou o desempenho do software, aprimorando as funções atuais ou adicionando novas funcionalidades.

Flexibilidade Facilidade de esforço para mudar a missão, as funções ou os dados do software para atender às necessidades e requisitos em constante mudança.

Portabilidade Facilidade de esforço para transportar software para outro ambiente e / ou plataforma.

Reutilização Facilidade de esforço para usar o software (ou seus componentes) em outros sistemas e aplicativos de software

Interoperabilidade	Esforço relativo necessário para acoplar o software em uma plataforma a outro software e / ou outra plataforma
Intraoperabilidade	Esforço requerido para comunicações entre componentes no mesmo sistema de software

Tabela 2. Fatores da qualidade do software (ASHRAFI, 2002).

2.3. Engenharia de Software

A maioria dos observadores concordaria que a Engenharia de Software começou como uma disciplina no final dos anos 1960 com as famosas “Conferências da OTAN”. Mas, na verdade, o software estava sendo desenvolvido e projetado por pelo menos 15 anos antes da primeira dessas conferências. As conferências, no entanto, deram a esta atividade um nome, “Engenharia de Software”, e um objetivo primordial, nomeadamente, lidar com a “Crise de Software” (OSTERWEIL, 2007).

Em 1977, a Divisão Federal de Sistemas da IBM Corporation estabeleceu um Programa de Engenharia de Software para criar um conjunto de práticas uniformes de software que tratam dos princípios de design, desenvolvimento e gerenciamento de software (MILLS, 1979).

A disciplina de Engenharia de Software combina tópicos de design resultantes de influências universitárias com o desenvolvimento de software e o conhecimento de gerenciamento da indústria (MILLS, 1999).

A Engenharia de Software se caracterizou como um processo que envolve o desenvolvimento de software, como uma atividade da criação de um sistema de software que, quando usado, resolve os problemas de alguns usuários, alavanca suas oportunidades ou satisfaz suas necessidades. (HICKEY, DAVIS, 2004).

Para conceituar melhor a Engenharia de Software encontramos na literatura algumas definições:

- A Engenharia de Software pode ser definida como o desenho e desenvolvimento sistemático de produtos de software e a gestão do processo de software. A engenharia de software é um conjunto crescente de disciplinas e procedimentos para o desenvolvimento e manutenção confiáveis de software (MILLS, 1999).

- A engenharia de software é o ramo da engenharia de sistemas preocupada com o desenvolvimento de sistemas intensivos em softwares grandes e complexos. Concentram-se nas metas do mundo real, serviços prestados por e restrições em tais sistemas; a especificação precisa da estrutura e comportamento do sistema e a implementação dessas especificações; as atividades necessárias para desenvolver uma garantia de que as especificações e as metas do mundo real foram atingidas; a evolução de tais sistemas ao longo do tempo e entre as famílias do sistema. Também se preocupa com os processos, métodos e ferramentas para o desenvolvimento de sistemas intensivos em software de maneira econômica e oportuna (FINKELSTEIN, KRAMER, 2000).
- A engenharia de software está entre a engenharia de sistemas e a integração de sistemas, aceitando a partir do sistema de engenharia os requisitos e recursos de software do sistema e fornecendo integração de sistemas com o software para atender a esses requisitos com esses recursos. Assim, o software total de um sistema é um produto conjunto de engenharia de sistemas e engenharia de software, que começa com um propósito de sistema definido e uma configuração definida de hardware (MILLS, 1999).

Desde a adoção do termo “Engenharia de Software”, surgido nas Conferências da OTAN em 1968, que provavelmente foi criado pelo Professor Fritz Bauer (RANDELL, 1996), até os nossos dias, a Engenharia de Software tem passado por inúmeras inovações, contudo é sabido que as inovações de engenharia de software levam um tempo surpreendentemente longo para percorrer até o uso diário (FINKELSTEIN, KRAMER, 2000).

2.3.1. Práticas de Engenharia de Software

As práticas de engenharia de software se aplicam ao longo de todo o ciclo de vida do software. A correspondência entre as práticas de engenharia de software e as atividades do ciclo de vida é mostrada na Tabela 3 (MILLS, 1999).

Atividades do Ciclo de Vida	Desenho			Desenvolvimento			Gestão	
	Design avançado	Projeto sistemático	Programação Sistemática	Gerenciamento de código	Engenharia de Integração	Revisões técnicas	Gestão de custos	Gestão de programas
Definição do sistema	●					●	●	●

Design de software	●	●	●	●	●	●	●
Desenvolvimento de software		●	●	●	●	●	●
Teste do sistema de software			●	●	●	●	●
Sistema e teste de aceitação			●		●	●	●
Suporte operacional			●			●	●

Tabela 3. Práticas e atividades relacionadas (MILLS, 1999)

A IEEE Computer Society e a Association for Computing Machinery (ACM) trabalharam em parceria um projeto iniciado em 1993 e concluído em 1999, para desenvolver um Guia para o Corpo de Conhecimento de Engenharia de Software (Software Engineering Body of Knowledge - SWEBOK). O projeto teve um caráter de literatura normativa. Um documento normativo prescreve o que um engenheiro deve fazer ao invés de descrever a variedade de coisas que o engenheiro pode fazer (BOURQUE et al, 1999).

2.3.2. Processos de Engenharia de Software

O processo de software define a maneira pela qual o desenvolvimento de software é organizado, gerenciado, medido, suportado e melhorado (independentemente do tipo de tecnologia de suporte explorada no desenvolvimento). Embora possam apresentar diferentes níveis de sofisticação no domínio de seus processos, todas as organizações envolvidas no desenvolvimento de software seguem um processo de algum tipo, implícito ou explícito, reproduzível, instrumentado, adaptável ou não (DERNIAME, KABA, WASTELL, 2000).

As empresas de software e as empresas em geral (que consideram cada vez mais o software como um fator competitivo fundamental) perceberam que a chave para uma entrega bem-sucedida (no prazo, dentro do orçamento e com a qualidade esperada) está no gerenciamento eficaz de seu processo de software (DERNIAME, KABA, WASTELL, 2000).

Um processo de engenharia de software é o modelo escolhido para gerenciar a criação de software desde o início do cliente inicial até o lançamento do produto acabado. Segundo WANG, KING (2000), o processo escolhido geralmente envolve técnicas, tais como:

- Análise,
- Desenho,
- Codificação,
- Teste, e
- Manutenção.

Vários modelos de processos diferentes existem e variam principalmente na frequência, aplicação e implementação das técnicas acima, como por exemplo, diferentes modelos de processos usam diferentes técnicas de análise, outros modelos tentam implementar a solução para um problema em uma abordagem definitiva, enquanto outros adotam uma abordagem iterativa pela qual versões cada vez maiores e mais completas do software são construídas a cada iteração do modelo de processo (WANG, KING, 2000).

2.3.3. Engenharia de Software Baseada em Componentes

A principal função da Engenharia de Software baseada em Componentes é lidar com o desenvolvimento de sistemas a partir de peças (componentes), o desenvolvimento de peças como entidades reutilizáveis e a manutenção e melhoria de sistemas, customizando e substituindo essas partes (CRNKOVIC, 2001).

2.3.4. Gestão do Conhecimento em Engenharia de Software

A necessidade de maior desenvolvimento das práticas de engenharia de software nas empresas aumenta a demanda por conhecimento sistemático e gerenciamento de habilidades em combinação com o uso ativo desse conhecimento para apoiar a tomada de decisões em todas as etapas do ciclo de vida do software (RUHE, 2003).

O capital intelectual de uma organização consiste em ativos tangíveis e intangíveis. Ativos tangíveis, que correspondem a conhecimento explícito documentado, podem variar para diferentes setores e aplicações, mas geralmente incluem manuais; diretórios; correspondência com (e informações sobre) clientes, fornecedores e subcontratados; inteligência concorrente; patentes; licenças; e conhecimento derivado de processos de trabalho (como propostas e artefatos de projeto). Os ativos intangíveis, que

correspondem a conhecimento explícito tácito e não documentado, consistem em habilidades, experiência e conhecimento das pessoas de uma organização (RUS, LINDVALL, 2002).

2.3.5. Engenharia de Requisitos

A engenharia de requisitos é o ramo da engenharia de software preocupado com as metas, funções e restrições do mundo real em sistemas de software. Também se preocupa com a relação desses fatores com as especificações precisas do comportamento do software e com sua evolução ao longo do tempo e entre as famílias de software (NUSEIBEH, EASTERBROOK, 2000).

O processo de requisitos é frequentemente descrito como uma série de atividades, tais como:

- **Elicitação:** aprendendo, descobrindo, extraindo, revelando ou descobrindo as necessidades de clientes, usuários e outras partes interessadas em potencial.
- **Análise:** análise das informações eliciadas pelas partes interessadas para gerar uma lista de requisitos candidatos, geralmente criando e analisando modelos de requisitos, com os objetivos de aumentar a compreensão e a busca por incompletude e inconsistência.
- **Triagem:** determinar qual subconjunto dos requisitos determinados por elicitación e análise é apropriado para ser endereçado em versões específicas de um sistema.
- **Especificação:** documentando o comportamento externo desejado de um sistema.
- **Verificação:** determinação da razoabilidade, consistência, completude, adequação e falta de defeitos em um conjunto de requisitos.

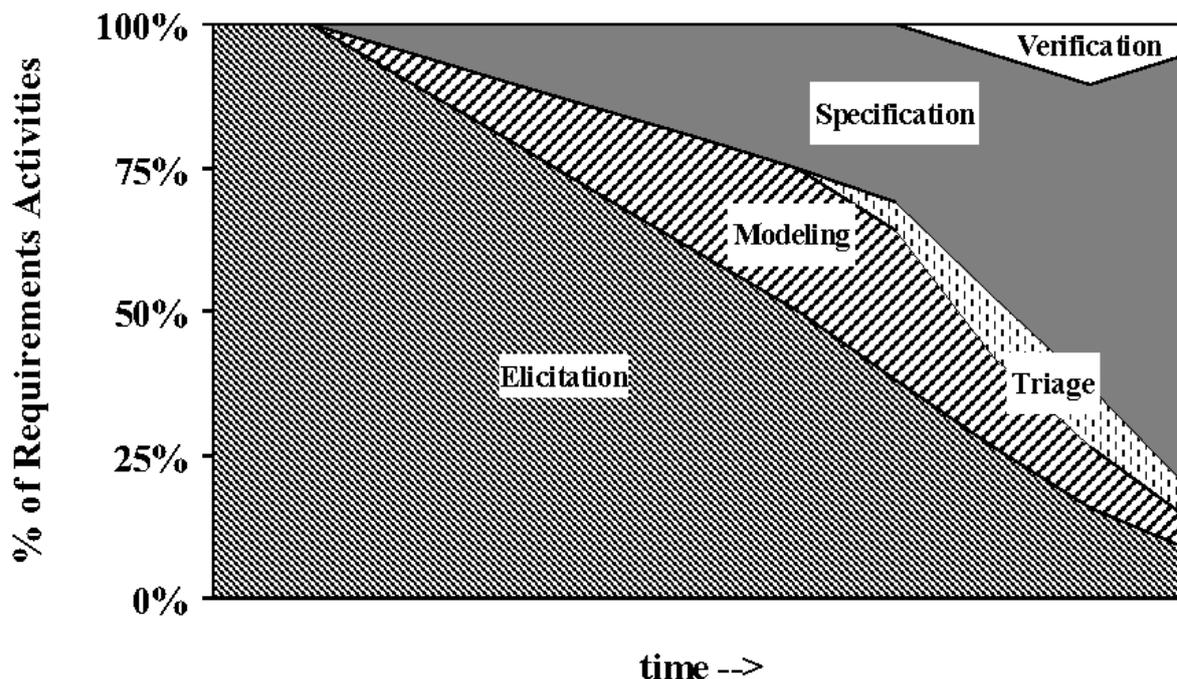


Figura 6. Modelo Paralelo do Processo de Requisitos (HICKEY, DAVIS, 2004).

A elicitação de requisitos tenta descobrir os requisitos e identificar os limites do sistema consultando as partes interessadas (por exemplo, clientes, desenvolvedores, usuários). Os limites do sistema definem o contexto do sistema. Compreender o domínio da aplicação, as necessidades de negócios, as restrições do sistema, as partes interessadas e o problema em si é essencial para obter uma compreensão do sistema a ser desenvolvido (PAETSCH et al, 2003).

A elicitação de requisitos talvez seja a atividade mais frequentemente considerada como o primeiro passo no processo de Engenharia de Requisitos. O termo “elicitação” é preferível a “capturar”, para evitar a sugestão de que os requisitos estão lá fora para serem coletados simplesmente fazendo as perguntas certas. As informações coletadas durante a elicitação de requisitos geralmente têm que ser interpretadas, analisadas, modeladas e validadas antes que o engenheiro de requisitos possa se sentir confiante de que um conjunto completo de requisitos de um sistema foi coletado. (NUSEIBEH, EASTERBROOK, 2000).

2.4. Metodologias de Desenvolvimento de Software

Nos anos 1970 difundiu-se internacionalmente uma das mais conhecidas metodologias de desenvolvimento de software, o modelo em cascata (*waterfall*), também conhecido como clássico ou linear. Nessa metodologia, largamente utilizada até o início da

década de 1990, o sistema é todo planejado e documentado, seguindo diversas fases, sequencialmente, antes de ser implementado. As atividades do processo de desenvolvimento são estruturadas em uma ordem fixa de fases (cascata), na qual as saídas de cada fase – geralmente documentos – após aprovadas tornam-se entradas para a próxima. Espera-se, no fim da última fase, que o produto de software esteja pronto, baseando-se na premissa de que o trabalho nas fases anteriores foi realizado de maneira correta e gerou os produtos que delas se esperava (BRASIL, 2013).

Embora tenha auxiliado a engenharia de software quando de seu surgimento, oferecendo uma forma disciplinada para a construção de soluções, com o advento de novas tecnologias, como microcomputadores utilizados em larga escala, o modelo em cascata passou a não atender às necessidades emergentes. Fomentaram-se, então, diversas críticas a esse modelo, como a impossibilidade de retornar de modo simples à execução de atividades em fases anteriores e a excessiva produção de documentação nas fases iniciais do projeto. Isso causava, respectivamente, uma elevação significativa do custo das mudanças no projeto e demora na disponibilização do software (BRASIL, 2013).

O Acórdão TCU 2314/2013 – Plenário, que teve o objetivo de conhecer a utilização de “métodos ágeis” nas contratações para desenvolvimento de software pela Administração Pública Federal, esclarece o seguinte sobre Metodologias de Desenvolvimento de Software:

De acordo com o dicionário Aurélio, no campo da literatura, metodologia é definida como o conjunto de técnicas e processos utilizados para ultrapassar a subjetividade do autor e atingir a obra literária. Já o dicionário Webster, no ramo da computação, conceitua metodologia como um conjunto organizado de procedimentos e guias documentados para a execução de uma ou mais fases do ciclo de vida do software.

Alinhando-se a essas definições, em engenharia de software, uma metodologia de desenvolvimento comumente é entendida como um conjunto estruturado de práticas que pode ser repetível durante o processo de produção do sistema ou, ainda, a forma de se utilizar um conjunto de práticas, métodos ou processos para se desenvolver ou manter um produto de software, de modo que se evite subjetividade na execução do trabalho. *O uso de metodologias visa a produtividade das equipes e a qualidade do produto* (grifo nosso).

As metodologias de desenvolvimento de software surgiram em um contexto tecnológico muito diferente do atual, baseado apenas em soluções para computadores de grande porte (mainframes) e terminais burros, época na qual o custo para se fazer alterações e correções nos sistemas era muito elevado. Isso se deu em decorrência do limitado acesso aos computadores e da inexistência de modernas ferramentas de apoio ao desenvolvimento.

2.4.1. Métodos de Desenvolvimento Ágeis

Em comparação com os processos de software tradicionais, o desenvolvimento ágil é menos centrado no documento e mais orientado ao código. Isso, no entanto, não é o ponto chave, mas sim um sintoma de duas diferenças mais profundas entre os dois (PAETSCH et al, 2003):

- Métodos ágeis são adaptativos e não preditivos. Com os métodos tradicionais, a maior parte do processo de software é planejada em detalhes por um grande período de tempo. Isso funciona bem se não houver muita mudança (ou seja, baixa rotatividade de requisitos) e o domínio do aplicativo e as tecnologias de software forem bem compreendidos pela equipe de desenvolvimento. Métodos ágeis foram desenvolvidos para adaptar e prosperar em mudanças frequentes.
- Os métodos ágeis são orientados para as pessoas e não para os processadores. Eles confiam na expertise, na competência e na colaboração direta das pessoas, e não em processos documentados e rigorosos para produzir software de alta qualidade.

Dentre os métodos de desenvolvimento ágeis, encontramos:

- Extreme Programming (XP) é baseado em valores de simplicidade, comunicação, feedback e coragem. Ele funciona reunindo toda a equipe na presença de práticas simples, com feedback suficiente para permitir que a equipe veja onde está. O XP combina técnicas antigas, testadas e comprovadas, de forma que elas se reforcem.



Figura 7. Diagrama do Método XP.

Fonte: <http://www.inf.ufpr.br/silvia/ES/SweES/pdf/AgeisA1.pdf>.

- O Scrum é um método para gerenciar o processo de desenvolvimento de sistemas, aplicando ideias sobre flexibilidade, adaptabilidade e produtividade da teoria de controle de processos industriais. Scrum se concentra em como uma equipe deve trabalhar em conjunto para produzir um trabalho de qualidade em um ambiente em mudança.

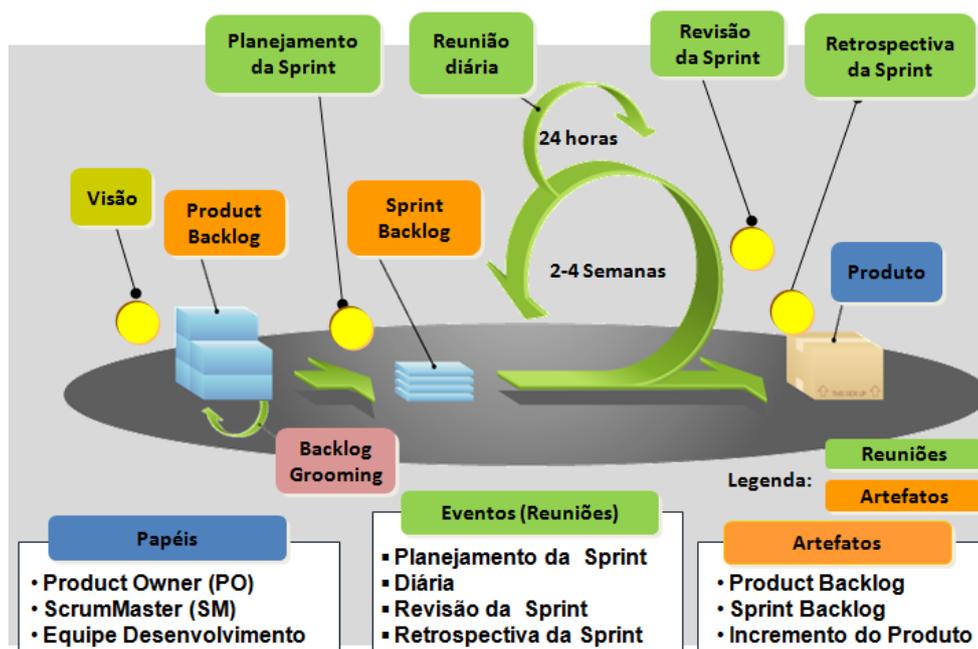


Figura 8. Diagrama do Método Scrum.

Fonte: <http://www.inf.ufpr.br/silvia/ES/SweES/pdf/AgeisA1.pdf>.

- A AM (Agile Modeling, ou Modelagem ágil) tem como ideia básica fornecer aos desenvolvedores uma diretriz de como construir modelos que ajudem a resolver problemas de design, mas não "sobre construir" esses modelos.
- As Metodologias de Cristal (The Crystal Methodologies) são uma família de metodologias diferentes das quais as metodologias apropriadas podem ser escolhidas para cada projeto. Os membros da família Crystal são indexados por cores diferentes para indicar o "peso": claro, amarelo, laranja, vermelho, magenta, azul, violeta, e podem ser adaptados para se adaptarem a diferentes circunstâncias.
- O FDD (Feature Driven Development, ou Desenvolvimento baseado em recursos), é um processo de iteração curta para desenvolvimento de software com foco na fase de projeto e construção, em vez de abranger todo o processo de desenvolvimento de software.
- O DSDM (Dynamic Systems Development Method, ou Método de Desenvolvimento de Sistemas Dinâmicos), fornece uma estrutura para o rápido desenvolvimento de aplicativos. As duas primeiras fases do DSDM são o estudo de viabilidade e o estudo de negócios. Durante essas duas fases, os requisitos básicos são elicitados. Outros requisitos são obtidos durante o processo de desenvolvimento.
- O ASD (Adaptive Software Development, ou Desenvolvimento adaptativo de software), fornece uma estrutura para o desenvolvimento iterativo de sistemas grandes e complexos. O método incentiva o desenvolvimento incremental e iterativo com prototipagem constante.

2.4.2. Métodos de Desenvolvimento Orientados a Serviços

A Computação Orientada a Serviços representa uma mudança de paradigma na engenharia de software, em que a principal abstração é a dos serviços, utilizada para suportar o desenvolvimento de aplicativos rápidos e de baixo custo por meio da composição de serviços. Embora a tecnologia e os padrões, como os serviços da Web, sejam importantes para alcançar a SOA (Service Oriented Architecture), é amplamente reconhecido que eles não são suficientes por conta própria (RAMOLLARI, DRANIDIS, SIMONS, 2007).

2.5. Análise

Os sistemas de software tornam-se cada vez mais críticos em todos os domínios da sociedade humana. Transporte, telecomunicações, entretenimento, saúde, militar, educação e assim por diante; A lista é quase infinita. Esses sistemas são usados não apenas por grandes corporações e governos, mas também por usuários individuais. Esse uso tão amplo de sistemas de informação resultou nesses sistemas contendo uma grande quantidade de informações críticas, que inevitavelmente precisam permanecer seguras. Portanto, embora seja importante garantir que os sistemas de software sejam desenvolvidos de acordo com as necessidades do usuário (requisitos funcionais), é igualmente importante garantir que esses sistemas sejam seguros.

3. Metodologia da Pesquisa

Este capítulo apresenta um detalhamento da metodologia realizado durante essa pesquisa, as fases da pesquisa, as etapas de aplicação da pesquisa, que inclui os aspectos metodológicos da pesquisa e os resultados a serem alcançados.

Minayo (2007, p. 44, Apud GERHARDT et al, 2009) define metodologia de forma abrangente e concomitante (...) a) como a discussão epistemológica sobre o “caminho do pensamento” que o tema ou o objeto de investigação requer; b) como a apresentação adequada e justificada dos métodos, técnicas e dos instrumentos operativos que devem ser utilizados para as buscas relativas às indagações da investigação; c) e como a “criatividade do pesquisador”, ou seja, a sua marca pessoal e específica na forma de articular teoria, métodos, achados experimentais, observacionais ou de qualquer outro tipo específico de resposta às indagações específicas.

3.1. Fases da Pesquisa

Atualmente é comum que um estudo prospectivo envolva o uso de múltiplos métodos ou técnicas, quantitativos e qualitativos, de modo a complementar as características diferentes de cada um, buscando compensar as possíveis deficiências trazidas pelo uso de técnicas ou métodos isolados. Uma vez que não faz sentido definir uma fórmula pronta para uma metodologia de prospecção, a escolha dos métodos e técnicas e seu uso dependem intrinsecamente de cada situação – considerados aspectos tais como especificidades da área de conhecimento, aplicação das tecnologias no contexto regional ou local, governamental ou empresarial, abrangência do exercício, horizonte temporal, custo, objetivos e condições subjacentes (SANTOS et al, 2004).

Para atender o objetivo geral desta pesquisa, serão utilizados os métodos e técnicas apresentados no Quadro 2, adaptados do modelo de WOHLIN & AURUM (2014).

Fase estratégica	1. Resultado da pesquisa	Pesquisa aplicada	
	2. Lógica da pesquisa	Pesquisa indutiva	
	3. Propósito da pesquisa	Descritiva	Explicativa

	4. Abordagem da pesquisa	Interpretativa	
Fase tática	5. Processo de pesquisa	Qualitativa	
	6. Metodologia da pesquisa	Cenários	
Fase operacional	7. Método de coleta de dados	Pesquisa ad hoc	Delphi
	8. Método de análise de dados	Análise temática	Hermenêutica

Quadro 4. Estrutura de tomada de decisão em pesquisa (adaptado de WOHLIN & AURUM, 2014).

Todas as etapas da pesquisa estão detalhadas a seguir, por cada uma das 3 (três) fases: estratégica, tática e operacional.

Fase Estratégica

1. Resultado da Pesquisa

A pesquisa **aplicada** é o tipo de pesquisa em que o pesquisador fornece uma solução para um problema específico aplicando conhecimento com o objetivo de melhorar a prática ou aplicação existente (Nunamaker et al. 1991; Collis e Hussey 2009 Apud WOHLIN & AURUM, 2014). A pesquisa aplicada não pode ficar sozinha; ele se baseia em pesquisa básica porque aplica o conhecimento científico a partir do conhecimento básico em uma prática existente (Bhattacharjee 2012, Apud WOHLIN & AURUM, 2014)

2. Lógica da Pesquisa

A pesquisa **indutiva** é baseada em argumentos indutivos, ela se move do específico para o geral. O pesquisador infere conceitos teóricos e padrões a partir de dados observados. O pesquisador começa com observações específicas, detecta padrões teóricos e desenvolve algumas conclusões ou teorias gerais (Basili, 1993, Apud WOHLIN & AURUM, 2014). Essa abordagem também é chamada de pesquisa de construção de teoria (Bhattacharjee 2012, Apud WOHLIN & AURUM, 2014).

3. Propósito da Pesquisa

A pesquisa **descritiva** é, como o próprio nome sugere, aplicada para descrever um fenômeno ou características de um problema. É mais focado que a pesquisa exploratória e vai além da pesquisa exploratória. As questões de pesquisa tendem a começar com "o que" ou "como", como ele pretende descrever os fenômenos (Collis e Hussey 2009, Apud WOHLIN & AURUM, 2014).

Pesquisas **explicativas** são aplicadas ao examinar a natureza de certas relações entre os elementos de um problema. As questões de pesquisa tendem a começar com o "porquê", uma vez que visa explicar os fenômenos. A resposta à pergunta “por que” pode envolver objetivos de pesquisa qualitativos e quantitativos, como o uso de entrevistas, pesquisas e análise de documentos (WOHLIN & AURUM, 2014).

4. Abordagem da Pesquisa

A pesquisa **interpretativa** visa compreender as atividades humanas em uma situação específica da perspectiva dos participantes, portanto enfatiza o contexto (Klein e Myers, 1999; Myers, 1997, Apud WOHLIN & AURUM, 2014). Ele rejeita a possibilidade de pesquisa “objetiva” e acredita que a pesquisa pode ser subjetiva. Ele assume que o comportamento é influenciado pelos significados que as pessoas atribuem aos eventos (Orlikowski e Baroudi, 1991, Apud WOHLIN & AURUM, 2014). Assume-se que a validade da pesquisa pode ser obtida por meio da coleta de dados qualitativos que sejam ricos e detalhados (Orlikowski e Baroudi, 1991), Apud WOHLIN & AURUM, 2014. Tende a usar métodos qualitativos, e entrevistas ou etnografias (Myers 1997, Apud WOHLIN & AURUM, 2014).

Fase Tática

5. Processo de Pesquisa

A pesquisa **qualitativa** é uma questão de investigação que visa estudar os fenômenos sociais e culturais. É realizado quando um pesquisador tem como objetivo compreender as perspectivas de seus sujeitos de pesquisa. A ideia principal é que, ao obter acesso às perspectivas de iniciados, os pesquisadores também podem obter acesso a novas formas de ver o mundo (Hannah e Lautsch, 2011, Apud WOHLIN & AURUM, 2014). Dados qualitativos referem-se a descrições verbais refletindo o mundo como visto pelos

participantes. A pesquisa qualitativa envolve o uso de coleta de dados qualitativos, como entrevistas, documentos escritos e observação do participante para entender e explicar os fenômenos sociais. Os métodos qualitativos são adequados para construir teoria, escrever descrições ricas, explicar relações e descrever grupos de normas, por ex. padrões, modelos e frameworks.

6. Metodologia da Pesquisa

Segundo HEIJDEN (2009), em projetos práticos de cenários, entre várias possibilidades, alguns são feitos para convergir e focar em torno de uma estratégia, e alguns visam desenvolver habilidade de antecipação. Alguns autores, de acordo com Mietzner e Reger (2004 Apud HEIJDEN, 2009), fazem distinção entre construção e planejamento de cenários. A construção de cenários leva em consideração as incertezas que cercam o futuro: avaliam e identificam possíveis resultados para futuros diferentes. Nesta concepção, a construção de cenários é a fundação necessária para o Planejamento de Cenários, uma metodologia de administração usada por gerentes para articular seus modelos mentais sobre o futuro e, desta forma, melhorar a tomada de decisão. Outros não distinguem cenários do próprio planejamento, eles os unem (JUNIOR et al, 2010).

Fase Operacional

7. Método de Coleta de Dados

O método de opinião de especialistas tem seus limites estabelecidos naquilo que as pessoas percebem como factível, de acordo com sua imaginação e crenças, e deve ser usada sempre que a informação não puder ser quantificada ou quando os dados históricos não estão disponíveis ou não são aplicáveis. Mesmo quando há dados históricos, a opinião de especialistas pode e deve ser usada como uma forma de complementar as informações obtidas e de captação de conhecimentos tácitos, sinais fracos e insights. Por isso, tais métodos são considerados qualitativos.

O método Delphi é recomendável quando não se dispõe de dados quantitativos, ou estes não podem ser projetados para o futuro com segurança, em face das expectativas de mudanças estruturais nos fatores determinantes das tendências futuras (WRIGHT & GIOVINAZZO, 2000).

8. Método de Análise de Dados

Análise temática: A análise temática é amplamente utilizada como técnica qualitativa de análise de dados em pesquisas empíricas de engenharia de software, uma vez que fornece uma compreensão mais profunda sobre o conteúdo dos dados. A análise temática geralmente envolve codificação aberta, onde os códigos são usados para organizar os temas, e deve envolver vários pesquisadores com temas sendo desenvolvidos usando discussões com entrevistados ou outros pesquisadores para captar as múltiplas perspectivas de várias pessoas (WOHLIN & AURUM, 2014).

Hermenêutica: é um método de análise de dados para compreensão e interpretação de dados qualitativos. O método requer a leitura de dados, compreendendo as pequenas partes dos dados que, em última análise, levam à compreensão de um todo maior. O círculo hermenêutico refere-se à dialética entre a compreensão do texto como um todo e a interpretação de sua parte (Myers 1995), a hermenêutica moderna envolve todos os tipos de dados, incluindo dados verbais e não verbais (Boell e Cecez-Kecmanovic, 2010) e é usada em muitas disciplinas que envolvem comportamento humano, incluindo informações, pesquisa de sistemas e engenharia de software. Boell e Cecez-Kecmanovic (2010) argumentam que a hermenêutica fornece uma estrutura para a revisão ad hoc da literatura, porque uma revisão ad hoc da literatura (em oposição a uma revisão sistemática da literatura) é um processo aberto de leitura e compreensão, é um processo iterativo e tem uma natureza cíclica (WOHLIN & AURUM, 2014).

No próximo capítulo apresentamos as etapas da pesquisa.

4. Resultados da Pesquisa

Tendo em vista que os processos de engenharia de software vêm se aperfeiçoando, desde a primeira vez que esse conceito foi apresentado nas conferências da OTAM em 1968, onde também foram apresentados problemas de escala, mas, no entanto, alguns desses problemas tem se perpetuado, inclusive tem surgido alguns outros novos problemas, como resultado da constante evolução da tecnologia, como o uso crescente da internet em escala comercial, a partir da década de 1990, o uso de aplicativos para celular, a internet das coisas, a robótica, a inteligência artificial, entre tantas outras, tem sido difícil para as empresas de desenvolvimento de software se planejarem adequadamente para se adaptarem a essas mudanças e para enfrentar esses problemas.

Certamente, tem havido um progresso gratificante em áreas como melhoria de linguagem de programação, abordagens de teste de software, suporte para fases de desenvolvimento de pré-codificação e métricas de software. Mas em cada área, as soluções atualmente disponíveis estão longe de ser adequadas para solucionar os problemas enfrentados pelos desenvolvedores de software (OSTERWEIL, 2007).

Os cenários criados como resultado final deste trabalho poderão ser usados como uma importante base de informações para as empresas envolvidas em desenvolvimento se preparem mais adequadamente em seus processos de desenvolvimento de software. Podendo também utilizar essas informações quando da elaboração e/ou da revisão do PDTI (Plano Diretor de Tecnologia da Informação) e do PETI (Planejamento Estratégico de Tecnologia da Informação).

A construção de cenários é feita a partir da aplicação de métodos e técnicas que capturam tendências passadas e atuais. A captura de tendências passadas e atuais; a antecipação e exploração de mudanças futuras e questões emergentes; previsões sobre o tamanho, escopo, identificação e restrição de vetores de mudança, avaliação da evolução de políticas e exploração de opções e suas implicações formam o escopo que serve de base para a construção de cenários.

Em resumo, este capítulo apresenta os dados obtidos com a aplicação do método de Pesquisa do Futuro Delphi, incluindo o processo de escolha de especialistas e a ferramenta adotada, e apresenta também os resultados da Construção de Cenários.

4.1. Etapas da Pesquisa

A Figura 9 apresenta as diversas etapas que foram executadas para atingir o objetivo desse trabalho de pesquisa.

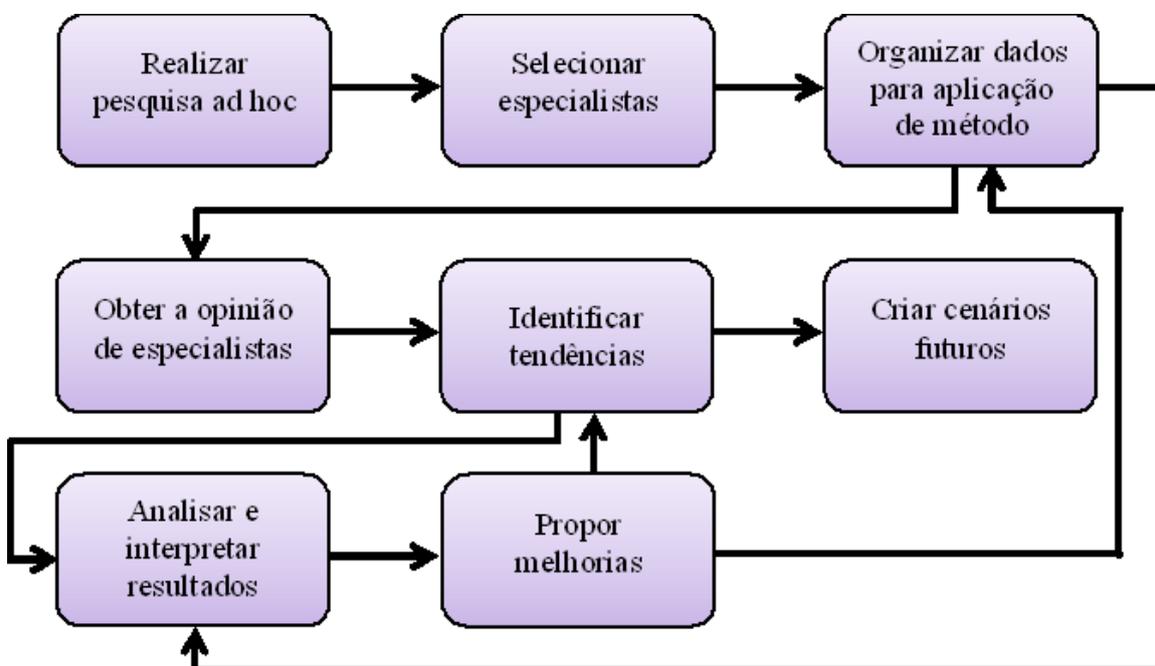


Figura 9. Etapas da pesquisa.

4.1.1. Realizar Pesquisa ad-hoc

A pesquisa ad-hoc foi realizada utilizando a ferramenta Google Acadêmico, por ser considerada uma boa opção para buscas mais simples concentradas em um único ponto de acesso. Criado em novembro de 2004, na versão em língua inglesa, o Google Acadêmico foi disponibilizado em português em janeiro de 2006. Ele fornece uma maneira simples de pesquisar literatura acadêmica de forma abrangente e é útil para pesquisar várias disciplinas e fontes em apenas um lugar: artigos revisados por especialistas, teses, livros, resumos e artigos de editoras acadêmicas, organizações profissionais, bibliotecas de pré-publicações, universidades e outras entidades acadêmicas (PUCCINI et al, 2015).

A pesquisa ad-hoc teve como argumento de pesquisa as palavras-chaves em inglês para a recuperação de dados "software engineering" and (*methods and methodologies and process and techniques*), considerando o intervalo de tempo entre os anos de 1999 até 2018, ou seja, os últimos 20 anos, incluindo apenas artigos e livros. Para refinar a pesquisa foram selecionados os 20 artigos/livros mais relevantes por ano segundo

os critérios de relevância da ferramenta, totalizando 400 trabalhos (artigos e livros). Como exemplo a Figura 10 mostra o resultado da pesquisa para o ano de 1999.

Destacamos que também utilizamos como argumento de pesquisa as palavras-chaves em inglês para a recuperação de dados "software engineering" *and (methods or methodologies or process or techniques)*, e que o resultado foi o mesmo do resultado anterior.

The screenshot shows the Google Acadêmico search interface. The search bar contains the query: "software engineering" and (methods or methodologies or process or techni. The results are filtered for the year 1999. The first result is a PDF titled "PuLSE: A Methodology to Develop Software Product Lines" by J. Bayer, O. Flege, P. Knauber, R. Laqua, and D. Muthig, published in SSR in 1999. The second result is a PDF titled "A comparative study of methods for hypermedia development" by N. Koch, published in the British journal of management in 1999. The third result is titled "From knowledge engineering to knowledge management" by N. Shadbolt and N. Milton, published in the British journal of management in 1999. The fourth result is titled "Overview of methodologies for building ontologies" by M. Fernández-López, published in the proceedings of the IJCAI-99 workshop on Ontologies and Problem-Solving Methods (KRR5) in Stockholm, Sweden in 1999.

Figura 10. Resultado da pesquisa para o ano de 1999 utilizando o Google Acadêmico.

Os artigos/livros foram catalogados em planilha Excel, por ano de publicação, contendo o título do artigo/livro, autor (es), e-mail dos autores, palavras-chave e resumo do trabalho.

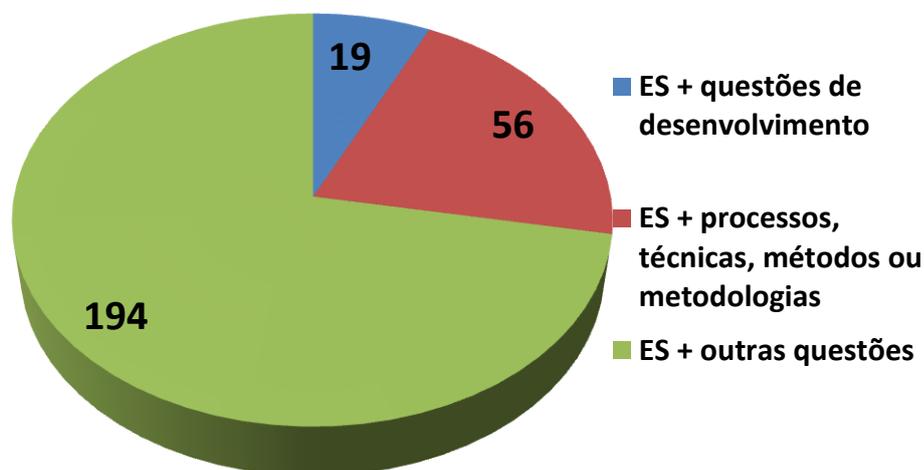


Gráfico 1. Incidências dos argumentos nos resumos dos artigos.

Após leitura e análise dos resumos dos artigos, 269 foram selecionados por se mostraram mais relevantes com o tema principal. Destes, 19 tratam sobre engenharia de software e questões sobre desenvolvimento de software, e 56 tratam sobre engenharia de software e pelo menos um dos outros argumentos de pesquisa: processos, técnicas, métodos ou metodologias, totalizando 75 artigos que foram lidos integralmente, conforme Gráfico 2.

Como resultado dessa pesquisa, encontramos os seguintes problemas nos processos de desenvolvimento de software da última década:

- A amplitude de aplicações na indústria enfatiza o desenvolvimento de software e resultou em uma diversidade de tecnologias de design, produtos de computador, linguagens de programação, ferramentas de software de suporte e requisitos de documentação. Mover-se de uma área de aplicação para outra pode exigir grandes ajustes, tanto pelos técnicos quanto pela gerência. Esses ajustes de consumo de tempo e energia introduzem mais diversidade e complicam ainda mais um processo já complexo (MILLS, 1999).
- O cenário geral é caracterizado por um lado, por uma crescente dependência comercial da confiabilidade da infraestrutura de software e, por outro lado, da rápida mudança e reconfiguração dos serviços corporativos, o que exige rápido desenvolvimento de software e mudanças frequentes nessa infraestrutura de software (FINKELSTEIN, KRAMER, 2000).
- A indústria de software ainda está lutando com a formidável tarefa de desenvolver aplicativos de software que atendam aos padrões de qualidade, pressão de tempo e restrições orçamentárias, apesar do rápido avanço da tecnologia. Os elementos de custo e tempo são quantitativos e, portanto, podem ser medidos e avaliados. A qualidade, por outro lado, é um conceito multidimensional, difícil de definir e medir (ASHRAFI, 2002).
- Lidar com o problema do envelhecimento de software. Sucintamente, o envelhecimento de software pode ser entendido como o acúmulo de erros (ocorrência de faltas internas) no software durante sua execução, levando a uma degradação progressiva do seu estado interno, bem como a exaustão de recursos essenciais para seu correto funcionamento, o que causa na maioria das vezes a falha do sistema (JUNIOR, 2006).

- Milhões de programas de computador tem que ser corrigidos, adaptados e aprimorados com o passar do tempo. O ônus de realizar essas atividades de “manutenção” absorve mais pessoas e mais recursos do que todo o trabalho aplicado à criação de novos softwares (PRESSMAN, MAXIM, 2015).
- À medida que a importância do software cresceu, a comunidade de software tentou continuamente desenvolver tecnologias que tornariam mais fácil, rápido e econômico criar e manter programas de computador de alta qualidade. Algumas dessas tecnologias são direcionadas a um domínio de aplicativo específico; outros se concentram em um domínio tecnológico; e ainda outros são de base ampla. No entanto, ainda precisamos desenvolver uma tecnologia de software que faça tudo, e a probabilidade de um surgimento no futuro seja pequena. (PRESSMAN, MAXIM, 2015).
- Retrospectivamente, podemos ver que a falta de conhecimento ou de práticas maduras de engenharia de software causou erros de software que, por sua vez, determinaram a perda de quantidades significativas de dinheiro nos melhores casos. O último caso proeminente aconteceu muito recentemente, em fevereiro de 2016, quando a espaçonave Japanese Hitomi foi interrompida por um problema causado por uma série de falhas de software (CASALEA et al, 2016).

4.1.2. Selecionar Especialistas

Para a seleção de especialistas fizemos a opção por especialistas no Brasil, com conhecimento em engenharia de software, com doutorado e com produção científica nos últimos 5 anos. A complexidade pela escolha de especialistas de qualquer nacionalidade consumiria muito tempo, tanto para escolha no universo de especialistas como para o contato no momento da pesquisa. Para a seleção fizemos pesquisa na Plataforma Lattes, buscando currículos com os seguintes critérios:

- Bases: Doutor
- Todas essas palavras: "engenharia de software"
- Qualquer uma dessas palavras: “futuro” ou “tendências”:
- Nacionalidade: brasileira

- Em preferências, com o período da produção a partir do ano 2015, que representa publicações nos últimos 5 (cinco) anos.

A seleção resultou num total de 327 especialistas, que foram catalogados em uma planilha Excel. Identificamos um problema inicial quando preparamos a base para estabelecer o contato com os especialistas, pois a plataforma Lattes não tem o e-mail pessoal no seu cadastro. Por esse motivo tivemos que pesquisar no Google Acadêmico publicações dos especialistas para identificar os e-mails de cada um deles individualmente. Conseguimos identificar 321 e-mails do total de 327 especialistas.

Os especialistas que responderam o questionário tinham o seguinte perfil acadêmico e profissional:

Especialista	Formação acadêmica/titulação	Atuação Profissional
A	Doutorado em Ciências da Computação pela Universidade Federal de Pernambuco	Professor Adjunto IV da Universidade Federal de Pernambuco, Brasil.
B	Doutorado em Desenvolvimento Sustentável do Trópico Úmido pela Universidade Federal do Pará	Professor Associado da Universidade Federal do Amazonas, Brasil. Instituto Nacional de Pesquisas da Amazônia, INPA, Brasil.
C	Doutorado em Administração pela Universidade Federal de Pernambuco	Professor do Ensino Básico, Tec. e Tecnológico da Universidade Federal Rural de Pernambuco.
D	Doutorado em Engenharia e Gestão do Conhecimento pela Universidade Federal de Santa Catarina	Professor do Instituto Federal Catarinense, Brasil.
E	Doutorado em Engenharia Elétrica pela Universidade de São Paulo	Divisão de Física Aplicada do Instituto de Estudos Avançados da USP. Docente Permanente do Instituto Tecnológico de Aeronáutica, ITA.
F	Doutorado em Ciências da Computação pela Universidade Federal de Pernambuco Pós-Doutorado, The Open University, OPEN, Inglaterra.	Gerente de Projetos de software do Centro de Estudos e Sistemas Avançados do Recife (CESAR) Vice-Coordenador do Mestrado Profissional em Engenharia de Software do Cesar.
G	Mestrado em Tecnologia em Saúde. Pontifícia Universidade Católica do Paraná, PUC/PR	Gerente de TI, Brasil. Gerência de projetos, análise, consultoria, criação e desenvolvimento de sistemas voltados a aplicações Internet e intranet.

Quadro 5. Perfil dos especialistas que responderam ao questionário.

4.1.3. Organizar Dados para Aplicação do Método

O Método de Estudos do Futuro Delphi de prospecção ou estudo prospectivo do futuro foi escolhido como ferramenta para obter a opinião dos especialistas. Para preparação dos dados antes da aplicação do método, se fez necessário seguir dois passos.

O primeiro passo desta etapa consistiu em identificar na literatura tendências e desafios que serviram de base para elaboração das perguntas iniciais do questionário, que foram as seguintes:

- A Internet é o ambiente de desenvolvimento.
- A Internet é a infraestrutura de arquitetura e execução.
- Os usuários são móveis, nômades e estão “sempre ligados”.
- A Internet é a infraestrutura básica de distribuição e de negócios.
- O desenvolvimento colaborativo está em crescimento.
- As metodologias ágeis estão sendo amplamente utilizadas.

O segundo passo desta etapa consistiu na elaboração do questionário exploratório, que foi aplicado aos especialistas selecionados.

O principal objetivo do questionário exploratório foi o de identificar as questões que mais poderão impactar no desenvolvimento de software no futuro, e não necessariamente identificar quais serão os métodos, as metodologias, os processos ou as técnicas.

Após a elaboração das perguntas iniciais, validamos o questionário com os Professores João Araújo, da Universidade Nova de Lisboa, e o Professor Hermano Perrelli de Moura, da Universidade Federal de Pernambuco, que analisaram e deram o parecer sobre as questões, que foram prontamente acatadas.

O questionário exploratório final, que foi aplicado aos especialistas, consta como Apêndice A deste trabalho.

4.1.4. Obter a Opinião dos Especialistas

O Método Delphi prevê que a opinião dos especialistas é obtida a partir da aplicação de questionários.

Ficou estabelecido que fossem aplicadas pelo menos duas rodadas de questionários. O Delphi pode ser dividido em duas fases: 1ª) exploratória, que compreende o primeiro questionário, e às vezes o segundo também, onde o objetivo é explorar completamente o assunto e prover informações adicionais; 2ª) de avaliação, é a fase de colher as visões dos especialistas, consenso ou oposição de ideias. No nosso caso foram aplicadas duas rodadas, a primeira exploratória e a segunda de avaliação.

O Questionário Exploratório foi elaborado de forma simples e fácil de responder, de forma que os respondentes pudessem fazer julgamentos a respeito do futuro com base no contexto planejado pela pesquisa sem necessidade de explicações adicionais.

Escolha da Ferramenta

Para a aplicação do questionário tivemos ainda que fazer a escolha de uma ferramenta. Fizemos pesquisa para identificar possíveis ferramentas candidatas para a aplicação do questionário usando o método Delphi Online.

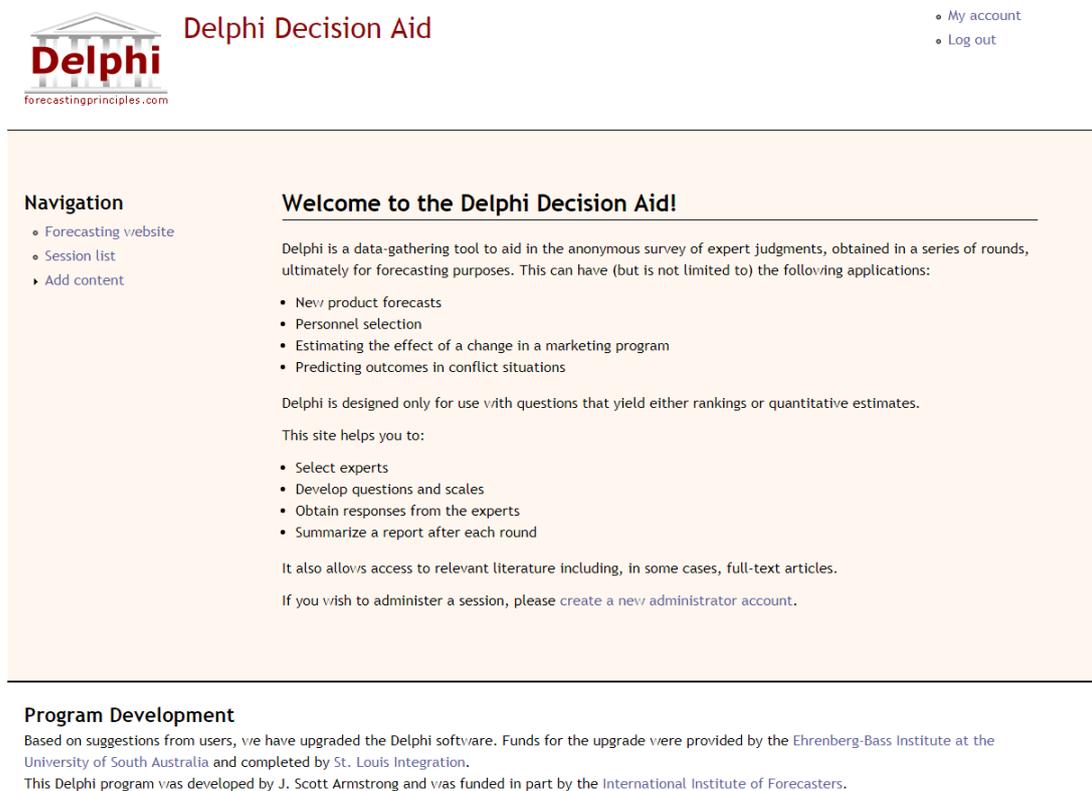
Identificamos e analisamos/testamos as seguintes ferramentas:

Ferramenta	Idiomas	Custo	Link	Conclusão
Real Time Delfoi	Finlandês	-	https://edelfoi.fi/	Não conseguimos testar, por conta do idioma.
Real Time Delphi - Estudos de Opinião Global	Inglês, português	Pago	https://realtimedelphi.org/index.php	Optamos por não testar, por ser pago e ter outras opções gratuitas.
Delphi Blue	Inglês	Open source	https://sourceforge.net/projects/delphiblue/	Não é atualizado desde 2014; dificuldades para instalar e testar.
TechCast Global	Inglês	Free trail	https://www.techcastglobal.com/join/	Muitas dificuldades para instalar e testar.
Software Delfos	Português	Open source	http://lhleonardo.github.io/delfos/	Ferramenta incompleta; o código da ferramenta não está concluído.
Delphi Decision Aid	Inglês	Sem custo	http://armstrong.wharton.upenn.edu/	Software completo. Foram feitos testes e avaliação.

Tabela 4. Comparativo das ferramentas para aplicação do Delphi online.

Após análise das ferramentas, optamos por testar a ferramenta Delphi Decision Aid. A Figura 11 apresenta a sua página inicial. Inicialmente a ferramenta se mostrou

satisfatória, tendo em vista suas características, e pelo fato de ser online, totalmente sem custo e de gerar relatórios e gráficos automaticamente a partir de sua aplicação.



Delphi
forecastingprinciples.com

Delphi Decision Aid

- My account
- Log out

Navigation

- [Forecasting website](#)
- [Session list](#)
- ▶ [Add content](#)

Welcome to the Delphi Decision Aid!

Delphi is a data-gathering tool to aid in the anonymous survey of expert judgments, obtained in a series of rounds, ultimately for forecasting purposes. This can have (but is not limited to) the following applications:

- New product forecasts
- Personnel selection
- Estimating the effect of a change in a marketing program
- Predicting outcomes in conflict situations

Delphi is designed only for use with questions that yield either rankings or quantitative estimates.

This site helps you to:

- Select experts
- Develop questions and scales
- Obtain responses from the experts
- Summarize a report after each round

It also allows access to relevant literature including, in some cases, full-text articles.

If you wish to administer a session, please [create a new administrator account](#).

Program Development

Based on suggestions from users, we have upgraded the Delphi software. Funds for the upgrade were provided by the Ehrenberg-Bass Institute at the University of South Australia and completed by St. Louis Integration.

This Delphi program was developed by J. Scott Armstrong and was funded in part by the International Institute of Forecasters.

Figura 11. Página inicial da ferramenta Delphi Decision Aid.

Desta forma criamos uma sessão para um teste inicial da ferramenta mapeando todos os passos que deveriam ser seguidos para sua aplicação (Figura 12). Em seguida geramos uma sessão com o questionário preliminar para testar o seu uso, onde três especialistas com conhecimento em engenharias de software foram convidados a colaborar testando não só a ferramenta bem como o questionário.

The screenshot shows the Delphi Decision Aid web interface. At the top left is the Delphi logo with the URL forecastingprinciples.com. At the top right are links for 'My account' and 'Log out'. The main content area is titled 'Round 1' and includes a 'Navigation' menu with options like 'Forecasting website', 'Session list', and 'Add content'. Below this is a 'View Round Edit' section with sub-links for 'Question designing', 'Expert Selection', 'Process monitoring', and 'Results review'. The session details are: 'Session name: Testando plataforma de Delphi Online', 'Round #: 1', and 'Round state: questions'. A 'View other rounds:' section lists actions like '[Round 1]', 'New question', 'Start round', 'Back to session list', and 'Log out'. The 'Round questions' section contains a table with two questions.

Title	Question type	Features	
Qual a data programada para sua defesa de dissertação/tese?	Open-ended	No opinion allowed: no	edit delete
Qual a probabilidade de você defender na data prevista?	Scaled	No opinion allowed: no Scale: 1 (Menos probabilidade de conseguir.) to 10 (Maior probabilidade de conseguir.)	edit delete

Figura 12. Sessão de teste da ferramenta Delphi Decision Aid.

Os três especialistas convidados tinham os seguintes perfis:

Especialista	Formação acadêmica/titulação	Atuação Profissional
A	Doutor em Ciência da Computação pela Universidade Federal de Pernambuco.	Professor da Universidade Federal Rural de Pernambuco
B	Mestre em Ciência da Computação pela Universidade Federal de Pernambuco.	Analista de Qualidade no Governo do Estado de Pernambuco
C	Doutor em Ciência da Computação pela Universidade Federal de Pernambuco.	Professor de Ciência da Computação.

Quadro 6. Perfil dos especialistas convidados para testar a ferramenta Delphi Decision Aid.

O resultado dos testes apresentado pelos especialistas não foi satisfatório, pelos seguintes motivos: primeiro, para participar do processo a ferramenta envia um “e-mail convite” muito genérico (Figura 13), não dando condições do aplicador do questionário elaborar um texto próprio para compor o corpo do e-mail, com o objetivo de esclarecer e sensibilizar os especialistas a participarem da aplicação do questionário. Segundo, a ferramenta não permite um prazo determinado para responder as questões, podendo gerar uma certa “acomodação” para os especialistas em responder o questionário. Terceiro, para

participar do processo é necessário que o especialista primeiro crie um cadastro com seus dados, para depois receber um segundo e-mail para responder as questões, o que pode gerar certo desconforto e criar uma barreira para a participação dos especialistas, sendo essa questão a mais relevante. Desta forma, confirmadas as dificuldades apresentadas, descartamos o seu uso.

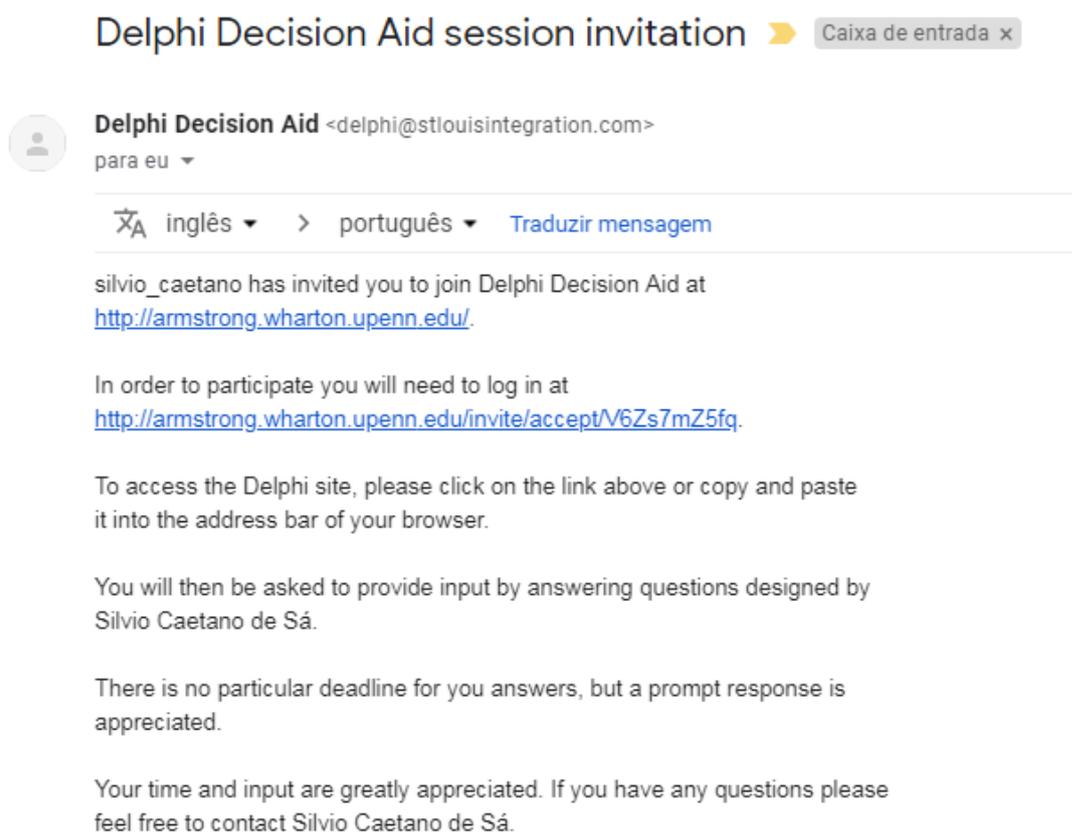


Figura 13. E-mail convite para aplicação da Ferramenta Delphi Decision Aid.

Tendo em vista o tempo exíguo, fizemos a opção pelo Google Forms, que é de simples utilização e gratuita, e de amplo conhecimento público.

O questionário utilizado na primeira rodada (Figura 14), elaborado no Google Forms, ficou ativo pelo período de 15 dias. Pode ser encontrado através do link: <https://docs.google.com/forms/d/1TH30gUVoqtgMbruDkDjSfleGUPv9fbuUNNhABgkdrd0/edit>.

← Pesquisa sobre "O futuro dos métodos de desenvolvimento de sof" ☆

PERGUNTAS RESPOSTAS 7

ENVIAR

Pesquisa acadêmica sobre O Futuro dos Métodos de Desenvolvimento de Software utilizando a técnica Delphi de Prospecção do Futuro.

Este trabalho faz parte de pesquisa de Mestrado Acadêmico em Ciência da Computação da Universidade Federal de Pernambuco, utilizando a técnica para Prospecção do Futuro Delphi, do Método de Opinião de Especialistas. Agradeço sua colaboração.

Por favor, responda com base em sua experiência pessoal e conhecimento em desenvolvimento de software, incluindo área de atuação e anos de experiência em engenharia de software, e em fábrica de software, desenvolvimento colaborativo, entre outros. Esta pesquisa estará disponível até o dia 31/05.

Descrição (opcional)

Nome

Figura 14. Página inicial do Questionário aplicado para a Primeira rodada.

O questionário utilizado na segunda rodada (Figura 15), elaborado no Google Forms, ficou ativo pelo período de 10 dias. Pode ser encontrado através do link: <https://docs.google.com/forms/d/e/1FAIpQLScVAmDMuJcxs08vID3hRc6NkmAgDx59jBbSQy9LkcLs7OBy4Q/viewform?vc=0&c=0&w=1>.

Convite para participar da 2ª rodada da pesquisa acadêmica sobre "O Futuro dos Métodos de Desenvolvimento de Software" utilizando a técnica Delphi de Prospecção do Futuro.

O formulário Convite para participar da 2ª rodada da pesquisa acadêmica sobre "O Futuro dos Métodos de Desenvolvimento de Software" utilizando a técnica Delphi de Prospecção do Futuro, não aceita mais respostas. Entre em contato com o proprietário do formulário se você achar que isso é um erro.

[Ver respostas anteriores](#)

[Retomar a coleta de respostas \(somente os editores do formulário podem ver este link\)](#)

Este formulário foi criado em Centro de Informática - UFPE. [Denunciar abuso](#) - [Termos de Serviço](#)

Figura 15. Página inicial do Questionário aplicado para a Segunda rodada.

Foram elaboradas 8 (oito) questões, conforme consta no Apêndice A - QUESTIONÁRIO EXPLORATÓRIO – 1ª RODADA, onde cada um dos especialistas deveria escrever sobre o seu entendimento para cada uma das questões. Todas as questões foram abertas.

Na primeira rodada encaminhamos 321 e-mails convidando os especialistas, que foram os que tínhamos conseguido e-mail. Destes, 29 e-mails voltaram com notificação de erro. Dois responderam que estavam aposentados e que não poderiam colaborar. Na primeira rodada 7 (sete) especialistas responderam o questionário. Na segunda rodada somente os 7 (sete) especialistas que participaram da primeira rodada foram convidados, destes, 4 (quatro) responderam o questionário.

Fizemos um esforço para incentivar a participação dos especialistas reencaminhando e-mails e até ligando para os que conseguimos telefone, para os que não responderam e que, pela avaliação do currículo, identificamos que seria importante ter a colaboração para enriquecer o nosso trabalho, mas não obtivemos sucesso.

Conclusões sobre o Uso do Método Delphi

O método Delphi comprova o seu potencial como possibilidade de articulação de abordagens qualitativas e quantitativas, somando o detalhamento e a subjetividade das informações da primeira abordagem com a objetivação e mensuração viabilizadas pela segunda.

No que tange as principais características do método, podem-se destacar a variedade de metodologias que o pesquisador pode utilizar para a validação e interpretação dos dados, sendo perfeitamente adequada para pesquisas futuras em engenharia de software.

O método Delphi alia a possibilidade de coletar dados de especialistas geograficamente dispersos, de áreas de conhecimento distintos, que tenham a mesma expertise do tema pesquisado, com a possibilidade destes mesmos especialistas chegarem a um consenso de opinião, aprofundando o conhecimento coletivo para a resolução de problemas futuros.

4.1.5. Identificar Tendências

Após aplicação do Questionário Exploratório, na primeira rodada, tendo sido consolidadas as respostas e submetidas à apreciação dos especialistas, através do Questionário Avaliativo, na segunda rodada, verificamos o percentual de concordância para cada questão, tendo o resultado mostrado no Quadro 7.

Questões	Descrição	Resposta Consolidada	Concordância
1	Considerando que a internet é a principal plataforma para desenvolvimento de software e para o acesso a dados e sistemas, os processos de engenharia de software atuais serão adequados para o software que será desenvolvido no futuro?	A internet é a principal plataforma para desenvolvimento de software e para o acesso a dados e sistemas, os processos de engenharia de software atuais não serão adequados para o software que será desenvolvido no futuro, havendo necessidade de adaptação nas metodologias, o uso de ferramentas de mais alto nível, além do que os processos precisam ser constantemente revisados.	75%
2	Se os processos de engenharia de software atuais não forem adequados para o software que será desenvolvido no futuro, como deverá ser a engenharia de software dentro desse cenário?	Os processos de engenharia de software atuais são adequados para o software que será desenvolvido no futuro, devendo, no entanto ser observadas as seguintes questões: ser baseado em software como serviços (SaaS), ser feito por pessoas altamente qualificadas para atender as exigências, a construção do software ser baseada em construção ágil, em modalidade incremental, sem necessidade de prévio desenho completo, Flexível e abrangente para adaptação, cada vez mais baseada no usuário em entregas de valor, simplicidade e com suporte de ferramentas para desenvolvimento distribuído.	100%
3	O desenvolvimento colaborativo de software está em crescimento. Como será o processo de desenvolvimento do software se o desenvolvimento colaborativo tiver um crescimento muito acentuado?	As metodologias atuais já contemplam, mas deve permitir que equipes geograficamente distantes possam avançar nos seus projetos, ser voltado para interfaces, necessitará de melhoria na integração e testes, de modo incremental e maior necessidade de ferramentas para dar suporte à gestão e comunicação entre as pessoas	100%
4	Se o desenvolvimento colaborativo tiver um	Transparência nas atividades para possibilitar um melhor controle, com boas práticas de comunicação, valorizar as	100%

Questões	Descrição	Resposta Consolidada	Concordância
	crescimento muito acentuado, o que será necessário a partir da evolução das empresas e das aplicações futuras?	ferramentas de colaboração e acesso remoto aos artefatos desenvolvidos pela equipe, melhoria dos requisitos, e fortalecer o gerenciamento do desenvolvimento.	
5	O software está cada vez mais especializado e independente na tomada de decisões. Imaginando que o software possa ser considerado um cidadão, como aconteceu com Sophia, como poderemos desenvolver software com a participação de um "software cidadão"?	Estabelecer claramente a responsabilidades para os integrantes do time, teria a capacidade de minerar fenômenos da sociedade através das redes sociais, tais como reclamações, passeatas, reivindicações, desastres, etc.; sendo assim, a aplicação de técnicas de social mining e machine learning serão imprescindíveis, utilizar o conhecimento agregado destes softwares, dedicação a padronização, dependerá das conquistas no campo da Inteligência Artificial, em especial a eficiência dos algoritmos Learning Machine.	75%
6	Quais os atributos de qualidade que são considerados atualmente e quais serão os atributos de qualidade relevantes no futuro?	Alta disponibilidade; segurança; velocidade em experimentação, inovação e mudanças. Atuais: segurança, rastreabilidade, maturidade, adequação, conformidade, funcionalidade, confiabilidade, usabilidade, eficiência, manutenibilidade, portabilidade e integração. Futuros: disponibilidade, escalonamento, tolerância a falhas, funcionalidade, confiabilidade, usabilidade, eficiência, aprendizado e auto ajuste, segurança, e performance.	100%
7	Quais as tendências atuais no desenvolvimento de software?	Maior modularidade e interação entre diferentes sistemas/serviços, software como serviço, app e integração, desenvolvimento ágil, experimentalismo, agregação incremental, construção de interfaces mais naturais e interativas.	100%
8	Quais as tendências futuras no desenvolvimento de software?	Conseguir acelerar entregas com valor realmente agregado reduzindo o tempo entre uma identificação de mudança/desenvolvimento e esta característica ir para o ambiente em produção o mais rápido possível, com o nível adequado de qualidade. Software baseado na Internet das Coisas em sistemas de tempo real com sensores. App, integração e interface com o usuário do futuro, Design Driven Development, Inteligência artificial embutida.	100%

Quadro 7. Respostas consolidadas do Questionário Avaliativo.

Para priorizar as questões como mais relevantes consideramos uma pontuação para cada resposta dada no Questionário Avaliativo, onde para cada questão equivale um valor pela sua importância, da seguinte forma:

- Concorda totalmente: vale 3 pontos.
- Concorda parcialmente: vale 2 pontos .
- Discorda parcialmente: vale 1 ponto.
- Demais questões: não pontuam.

Pontos	Respostas	Quantidade de respostas por questão							
		1	2	3	4	5	6	7	8
3	Concorda totalmente	2	2	3	4	2	4	4	3
2	Concorda parcialmente	1	2	1	-	1	-	-	1
1	Discorda parcialmente	1	-	-	-	-	-	-	-
0	Discorda totalmente	-	-	-	-	-	-	-	-
0	Não sei/não quero opinar	-	-	-	-	1	-	-	-
0	Outro: (alguma explicação ou comentário adicional)	-	-	-	-	-	-	-	-
PONTUAÇÃO		9	10	11	12	8	12	12	11

Tabela 5. Pontuação das questões a partir do quantitativo de respostas do Questionário Avaliativo.

A partir dos temas das 8 (oito) questões do questionário, identificamos 6 variáveis chaves, que serão usadas para determinar o impacto potencial e a probabilidade de desenvolvimento, que são as seguintes:

	C1	C2	C3	C4	C5	C6
Variáveis chaves	Internet como plataforma para desenvolvimento	Processos de engenharia de software	Desenvolvimento colaborativo de software	Software mais especializado e independente	Atributos de qualidade	Tendências no desenvolvimento de software
Questões associadas	1	2	3,4	5	6	7,8

Quadro 8. Variáveis chaves identificadas a partir das respostas dos questionários.

A partir da pontuação das questões, que foram associadas a cada uma das variáveis, adicionamos a pontuação para cada variável chave, e para as variáveis que tiveram duas questões associadas somamos a pontuação de ambas e dividimos por 2, tendo como resultado o seguinte:

	C1	C2	C3	C4	C5	C6
Variáveis chave	Internet como plataforma para desenvolvimento	Processos de engenharia de software	Desenvolvimento colaborativo de software	Software mais especializado e independente	Atributos de qualidade	Tendências no desenvolvimento de software
Questões associadas	1	2	3 e 4	5	6	7 e 8
Pontuação	9	10	11,5	8	12	11,5

Quadro 9. Totalização da pontuação para as variáveis chaves.

O próximo passo foi classificar as variáveis de acordo com a pontuação em **impacto potencial**, como ALTA as variáveis que tiveram entre 11 e 12 pontos, como MÉDIA as variáveis que tiveram pontuação entre 9 e 10, e como BAIXA as variáveis que tiveram pontuação abaixo de 9. A classificação de probabilidade de desenvolvimento levou em consideração a confirmação ou negação de cada resposta. Caso alguma questão fosse classificada como BAIXO impacto e BAIXA probabilidade seria automaticamente descartada.

Probabilidade de Desenvolvimento	Alta	C4		C6
	Média		C1	C3
	Baixa			C2, C5
		Baixa	Média	Alta

Impacto potencial

Quadro 10. Demonstrativo da probabilidade de desenvolvimento e de impacto potencial das variáveis chaves.

4.1.6. Analisar e Interpretar Resultados

As respostas se mostraram expressivas para o trabalho, conforme podemos observar no Gráfico 3. De 56 respostas possíveis (7 especialistas X 8 respostas do

questionário), descartando as respostas “Não sei/não posso responder”, “Não entendi a pergunta” e as parcialmente respondidas, tivemos um total de 42 respostas com informações válidas, o que representa 75% do total possível.

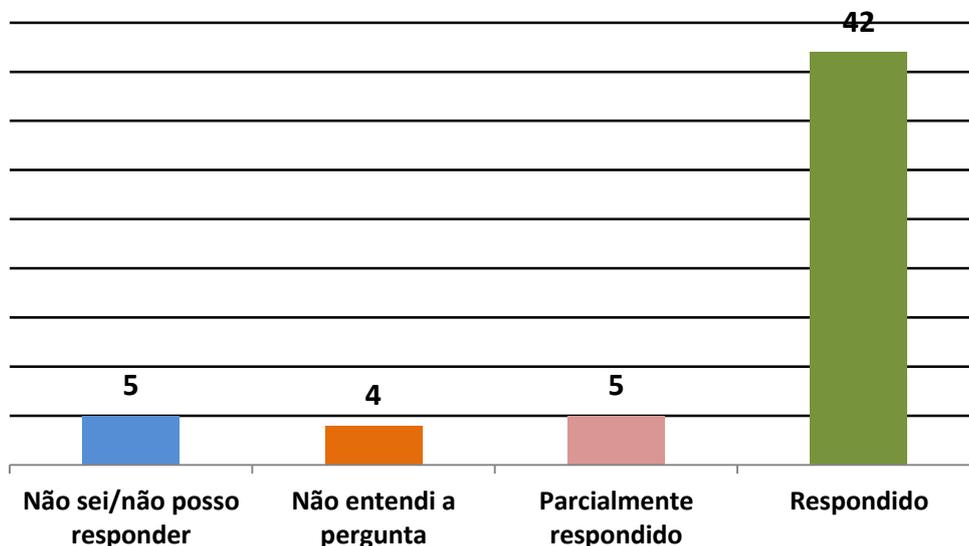


Gráfico 2. Distribuição das respostas para as perguntas do questionário.

Nenhuma questão teve mais de uma resposta para os itens “Não sei/não posso responder” e “Não entendi a pergunta”. Mas 3 questões tiveram “Não sei/não posso responder” e “Não entendi a pergunta” cumulativamente, ou seja, essas 3 questões só tiveram 5 respostas escritas cada.

Para consolidar as respostas seguimos 3 (três) passos: a) listamos sequencialmente as respostas brutas sem identificar os respondentes; b) verificamos as convergências nas respostas para cada uma das questões e relacionamos os principais itens; e c) elaboramos uma resposta no formato de frase única contemplando todos os itens coletados e estruturando as respostas para melhor entendimento.

Como exemplo demonstramos abaixo os 3 (três) passos seguidos para consolidar as respostas da Questão 4: “*Se o desenvolvimento colaborativo tiver um crescimento muito acentuado, o que será necessário a partir da evolução das empresas e das aplicações futuras?*”:

a) **Passo 1** – as respostas brutas foram listadas;

Transparência nas atividades para possibilitar um melhor controle, bem como boas práticas de comunicação.

Os processos deveriam valorizar as ferramentas de colaboração e acesso remoto aos artefatos desenvolvidos pela equipe.

Pessoal qualificado em interfaces

Melhoria na comunicação e requisitos.

Fortalecimento no gerenciamento do desenvolvimento

Não sei responder

Não entendi a pergunta.

b) **Passo 2** – as convergências das respostas foram identificadas;

Não entendi/sei responder – 2

Transparência nas atividades, boas práticas/melhoria na comunicação, ferramentas de colaboração acesso remoto, pessoal qualificado em interfaces, fortalecimento no gerenciamento, melhoria nos requisitos.

c) **Passo 3** – elaboração de frase única a partir dos itens de convergência;

“Transparência nas atividades para possibilitar um melhor controle, com boas práticas de comunicação, valorizar as ferramentas de colaboração e acesso remoto aos artefatos desenvolvidos pela equipe, melhoria dos requisitos, e fortalecer o gerenciamento do desenvolvimento.”

Todas as respostas consolidadas constam do Apêndice B – Questionário de Avaliação que foi aplicado na segunda rodada. Para sua aplicação utilizamos a escala Likert de 5 (cinco) pontos, para que os especialistas pudessem emitir o seu grau de concordância com cada afirmação. Os 5 (cinco) pontos foram os seguintes:

- a) Concorda totalmente,
- b) Concorda parcialmente,
- c) Discorda parcialmente,
- d) Discorda totalmente e
- e) Não sei/não quero opinar.

Ainda foi dada a possibilidade para os especialistas emitirem um complemento para a sua resposta, em um campo aberto, mas nenhum deles utilizou essa opção.

4.1.7. Propor Melhorias

Como resultado da análise e interpretação de dados, para os itens que apresentarem divergências, podem ser inclusive criadas novas questões que serão apresentadas na próxima rodada, o que, no nosso caso em particular, isso não aconteceu.

Na próxima rodada os especialistas são solicitados a indicarem se concordam com a opinião da maioria, estabelecendo assim o consenso.

4.1.8. Criar Cenários Futuros

O resultado final da aplicação do método Delphi é a base fundamental para a criação de cenários possíveis para o que mais influenciará os métodos de desenvolvimento de software no futuro. Esse processo está detalhado na Seção 4.1.5 (Identificar Tendências).

Para a fase de construção dos cenários foi escolhida uma equipe para colaborar de maneira produtiva, que HEIJDEN (2009, pág. 254) considera que deve possuir as seguintes características:

- Destacar-se dos outros pela agilidade de sua mente;
- Mestres das suas áreas;
- Curiosidade aguda e insaciável;
- Geradores de ideias;
- Acessíveis;
- Atenção constante a maneira como o mundo funciona;
- Observação aguda;
- Percepção singular;
- Estica os limites;
- Já vivendo no futuro.

Para compor a equipe de construção dos cenários convidamos dois profissionais de engenharia de software com pelo menos 15 anos de experiência, que também estão envolvidos em pesquisa de engenharia de software, e que ao mesmo tempo se identificam com muitas dessas características, possuindo os seguintes perfis:

Profissional	Formação acadêmica/titulação	Atuação Profissional
A	Mestre em Engenharia de Software pelo Cesar School. Doutorando em Ciência da Computação pela Universidade Federal de Pernambuco.	19 anos de experiência em Engenharia de Software. 5 anos de pesquisa em Engenharia de Software.
B	Mestre em Ciência da Computação pela Universidade Federal de Pernambuco.	Analista de Arquitetura de TIC do Tribunal de Justiça de PE. Professor Universitário de Sistemas de Informação.

Quadro 11. Perfil dos profissionais que participaram da elaboração dos cenários.

O trabalho da equipe foi a elaboração de diagramas de influência, que começa com a articulação de uma série de variáveis fundamentais, consideradas importantes para a situação considerada HEIJDEN (2009). De posse das variáveis fundamentais, que foram encontradas no Seção 4.1.5. Identificar tendências, os participantes identificaram a relação causal existente entre as variáveis, ou seja, qual variável ou conjunto de variáveis influencia as demais e em que intensidade, cujo objetivo é identificar os direcionadores de cenários e o grau de dependência.

Os trabalhos foram conduzidos conforme as atividades descritas no Quadro 12.

Atividade	Pesquisador	Equipe
Listar padrões e tendências	Não	Sim
Elaborar diagramas de influência	Não	Sim
Listar as forças motrizes	Sim	Sim
Classificar as forças motrizes	Sim	Sim

Quadro 12. Atividades da equipe de elaboração dos Cenários.

Listar Padrões e Tendências

O resultado desta etapa do trabalho está no Apêndice C, tendo sido identificados 4 (quatro) Cenários:

1. **Interfaces mais naturais**
2. **IoT – Internet das Coisas**
3. **Transparência das atividades**
4. **Construção ágil**

Elaborar Diagramas de Influência

Com base nestes cenários foram elaborados 4 diagramas de influencia, um para cada cenário, levando em conta as relações causais entre as variáveis, considerando apenas os elementos relevantes para cada contexto, que foram os seguintes:

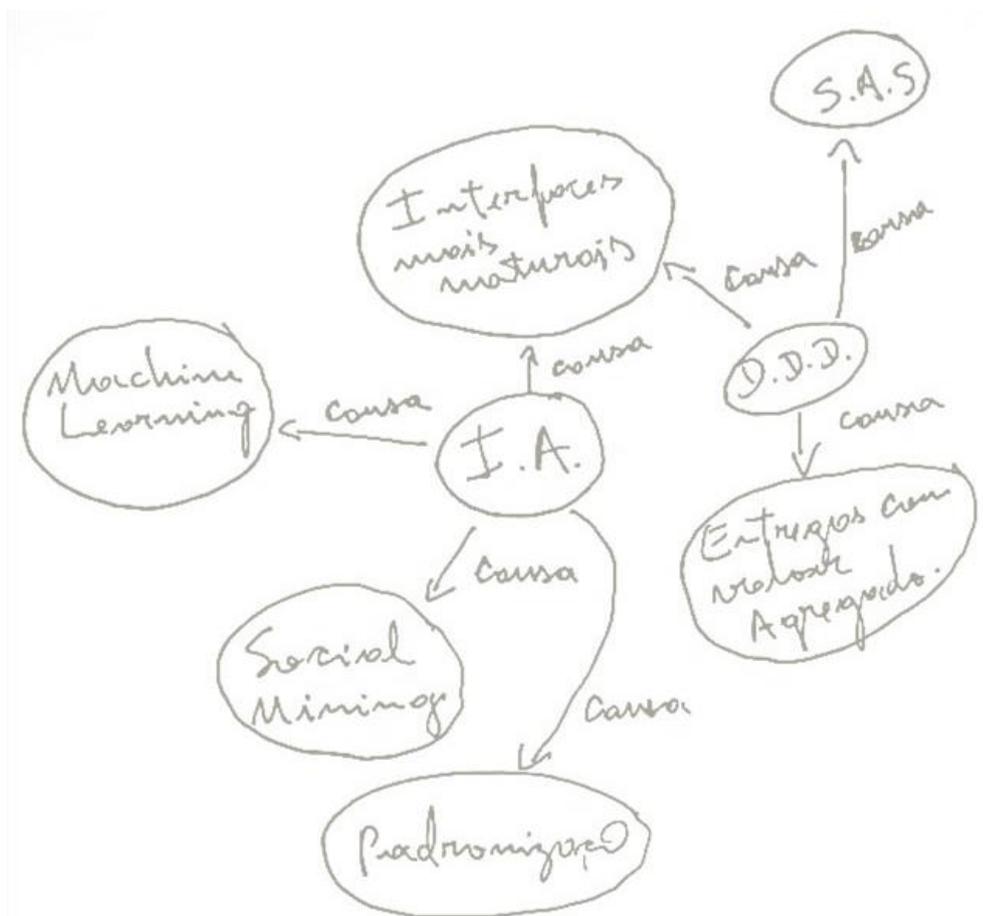


Figura 16. Diagrama de influência 1 – Interfaces mais naturais.

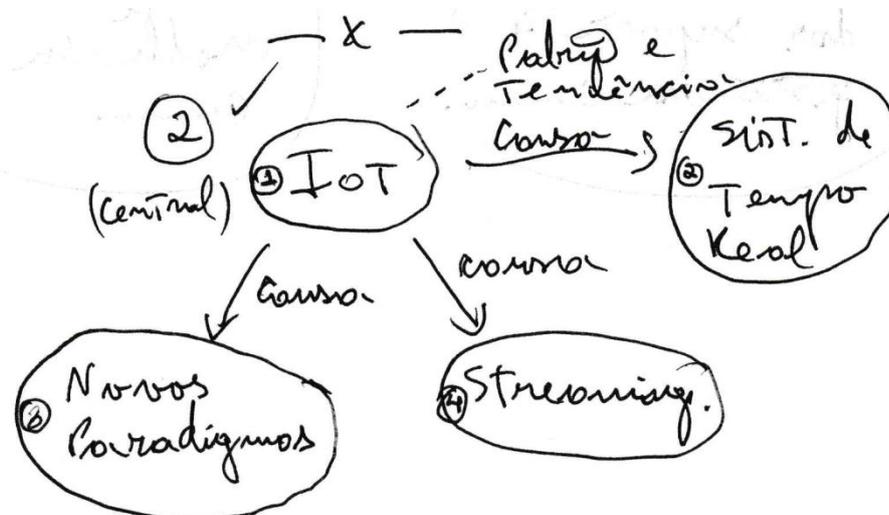


Figura 17. Diagrama de influência 2 – IoT – Internet das Coisas.

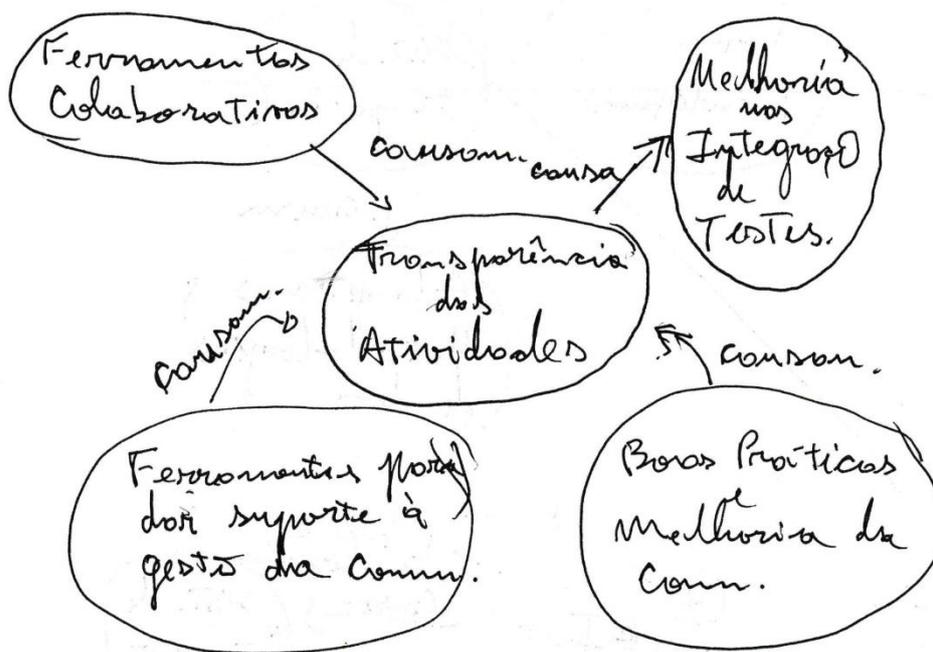


Figura 18. Diagrama de influência 3 – Transparência das atividades.



Figura 19. Diagrama de influência 4 – Construção ágil.

Listar as Forças Motrizes

Os diagramas servem para identificar as forças motrizes, que significa passar de eventos para o nível de padrões e tendências, e deste para o de estrutura, a fim de identificar as forças que afetam a situação fundamentalmente. Desta forma podemos identificar para cada diagrama os eventos de mais forte influência sobre os demais que devem ser observados, quais sejam:

Diagrama de Influência 1: Interfaces mais Naturais. Destacam-se os eventos de **IA** (Inteligência Artificial) e **DDD** (Design Driven Development), como os que mais influenciam os demais eventos, sendo o evento de **IA** o mais influente para o contexto.

Diagrama de Influência 2: Internet das Coisas. *IoT* foi considerado tanto padrão como tendência, e de influência total sobre os demais eventos. É interessante também observar que *novos paradigmas* devem surgir, sendo, portanto, uma tendência que deve ser melhor observada para o futuro.

Diagrama de Influência 3: Transparência das Atividades. Observamos que a *Transparência das Atividades* recebe influência direta dos demais eventos, que são *ferramentas colaborativas, ferramentas para dar suporte à gestão da comunicação e boas práticas e melhoria da comunicação*, que, por sua vez, influencia a *melhoria nas integrações de testes*.

Diagrama de Influência 4 – Construção Ágil. A *Construção ágil* afeta diretamente os demais eventos, e só recebe influência de mudanças. Ficaram em destaque dois eventos em que a *maturidade*, que por sua vez influencia diretamente a *confiabilidade*, mas que não recebe ou influencia diretamente a *construção ágil*.

Os cenários foram estruturados usando a abordagem matricial. Ela é especificamente apropriada em situações de incerteza considerável, em que poucas dimensões delineiam boa parte do que sabemos sobre a situação no futuro (HEIJDEN, 2009).

Classificar as Forças Motrizes

Utilizamos a Abordagem Matricial para criação dos cenários, classificando-as pelo seu grau de impacto sobre a ideia do negócio e seu nível de incerteza. As variáveis que devem ser observadas são as de maior impacto sobre a ideia do negócio e com menor

previsibilidade. Sendo assim 3 (três) variáveis que mais se destacam nos Cenários são: **inteligência Artificial**, **DDD** (Design Driven Development) e **Mudanças**.



Figura 20. Espaço de classificação das forças motrizes - Interfaces mais naturais.

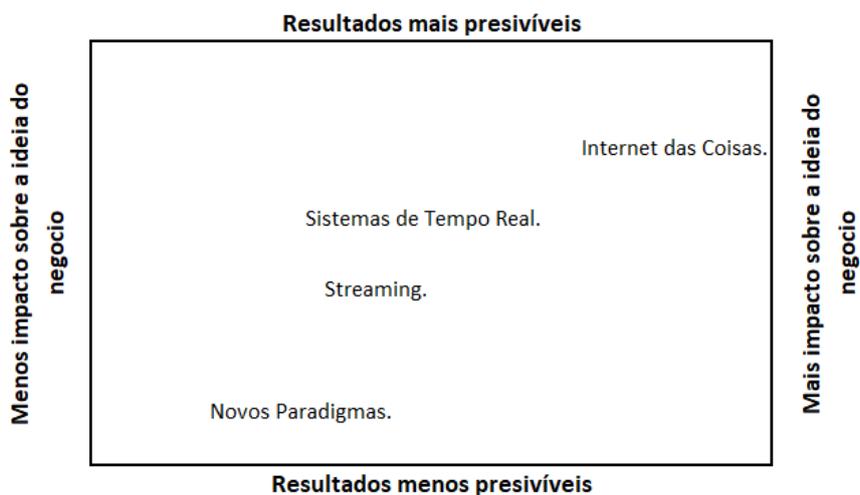


Figura 21. Espaço de classificação das forças motrizes – Internet das Coisas

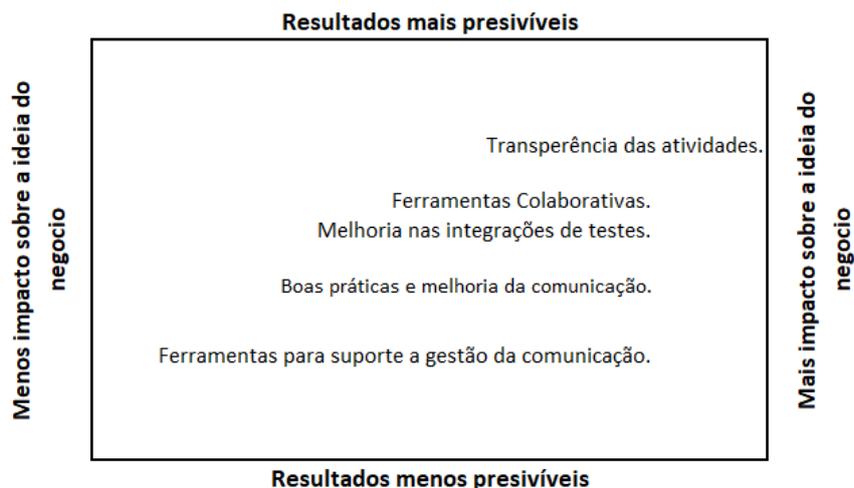


Figura 22. Espaço de classificação das forças motrizes – Transparência das atividades.



Figura 23. Espaço de classificação das forças motrizes – Construção Ágil.

4.2. Discussão dos resultados

O Método de Construção de Cenários, por suas características, necessita seguir certos passos para coletar as informações iniciais que servirão de base para identificação de variáveis chaves e consequente construção dos cenários. O primeiro passos envolve um processo de seleção de membros para compor a equipe que irá participar dessa construção, envolvendo questões como: amplo conhecimento sobre o assunto, poder de influência sobre a situação, desejo de influenciar, além de serem encorajados a se preparar, estudando sobre o tema, entre outras (HEIJDEN, 2009). Além disso, é necessário também criar o design do projeto, elaborar agenda de atividades, escolher um ambiente próprio para reuniões, elaborar cronograma, entre outras atividades. Tudo isso são fatores que apoiam o

processo de coleta de informações, e consequente identificação de variáveis chaves, que são básicas para o sucesso da construção de Cenários.

O Método de Estudos do Futuro Delphi se mostrou uma importante alternativa para conseguir coletar dados e identificar variáveis relevantes que são aplicadas no Método de Construção de Cenários. Para o nosso trabalho o uso do Delphi permitiu ultrapassar todas as atividades envolvidas na identificação de variáveis chaves, com bem menos recursos do que o habitual. Destacamos que nem sempre o método Delphi poderá ser utilizado para apoiar a construção de cenários, e só conseguimos ultrapassar essa etapa pelas peculiaridades do nosso objeto de pesquisa, no caso os métodos de Engenharia de Software. Para outros objetos de pesquisa o uso do Método Delphi deverá ser melhor avaliado.

5. Conclusões e Trabalhos Futuros

Nas conferências da OTAN, em 1968, alguns problemas foram apresentados, e diziam respeito basicamente a três pontos, gastar mais tempo do que o previsto, custar mais do que o orçado e não atender o que tinha sido requerido, e que foi chamada de “crise do software”. Nessas mesmas conferências também foi usado pela primeira vez o termo “Engenharia de Software”, com o objetivo de sistematizar e padronizar os processos de desenvolvimento de software.

Passados mais de 50 anos, os problemas, apesar de terem sido minimizados, ainda continuam a acontecer, e o que é pior, novos problemas têm surgido, pois o crescente desenvolvimento de novas tecnologias tem requerido software cada vez mais especializado, e a tendência é que cresçam exponencialmente. Além disso, a quantidade gigantesca de informações geradas que precisam de tratamento para a guarda e o acesso, o uso do software por praticamente todas as áreas do conhecimento humano, tornam a tarefa de desenvolver software com qualidade e segurança cada vez mais difícil.

5.1. Contribuições

O objetivo geral deste trabalho foi alcançado tendo em vista a construção de 4 (quatro) cenários que apresentam perspectivas para ajudar a resolver os vários problemas enfrentados na produção do software, buscando tendências futuras para que as empresas de desenvolvimento de software possam se antecipar preparando-se para criar ou melhorar os processos de desenvolvimento de software visando o futuro.

Como previsto nos objetivos específicos, após investigação da literatura, como resultado de pesquisa ad-hoc, alguns problemas nos métodos de desenvolvimento de software relacionados à engenharia de software foram detectados, além dos que se apresentaram como causadores da Crise do Software, e mais recentemente da Crise do Software 2.0, que estão relatados no item 4.1.1. Realizar Pesquisa ad-hoc.

Como resultado da aplicação do Método de Estudo de Futuro Delphi, que se caracteriza pela consulta aos especialistas, que no nosso caso específico foram da área de Engenharia de Software, através de duas rodadas de questionários, onde os mesmos deram sua visão sobre algumas questões, após chegarem ao consenso, as respostas foram

apresentadas como **tendências** que poderão afetar os processos de desenvolvimento de software no futuro.

Além disso, foram criados 4 (quatro) cenários, a partir das tendências, resultado da aplicação do Método de Estudo de Futuro Delphi, que indicam poderão ajudar empresas ou indivíduos a se planejarem melhor na melhoria dos processos de desenvolvimento de software no futuro.

5.2. Trabalhos Futuros

- Ampliar a aplicação do Método Delphi para especialistas não só do Brasil, visando a perspectiva de surgirem novas tendências;
- Fazer uma pesquisa mais aprofundada para verificar a incidência de artigos sobre métodos, metodologias processos, ferramentas e técnicas de engenharia de software nas últimas décadas;
- Aplicar o método de detecção de Sinais Fracos, identificando tendências ainda imperceptíveis pelos especialistas.
- Criar um aplicativo para aplicação do Método Delphi.

Referências

- AALTONEN, Mika; BARTH, Theodor. How **Do We Make Sense of the Future?** Journal of Futures Studies, May 2005, 9(4): 45 – 60.
- ABNT; SEBRAE. **Guia de implementação: Desenvolvimento de software para pequenas organizações.** Associação Brasileira de Normas Técnicas, Serviço Brasileiro de Apoio às Micro e Pequenas Empresas. Rio de Janeiro. 2012.
- AMER M; DAIM T. U; JETTER A. **A review of scenario planning.** In Futures, February 2013.
- ASHRAFI, Noushin. **The impact of software process improvement on quality: in theory and practice.** Department of MSIS College of Management, University of Massachusetts. Elsevier Science B.V. 2002.
- BOEHM, B. The Future of Software Processes. In: Li M., Boehm B., Osterweil L.J. (eds) **Unifying the Software Process Spectrum.** Lecture Notes in Computer Science, vol 3840. Springer, Berlin, Heidelberg. SPW 2005.
- BOURQUE, Pierre; DUPUIS, Robert; ABRAN, Alain; MOORE, James W.; TRIPP, Leonard. **The Guide to the Software Engineering - Body of Knowledge.** 0740-7459/99/IEEE. 1999.
- BRASIL. TCU – Tribunal de Contas da União. **Utilização de “métodos ágeis” nas contratações para desenvolvimento de software.** Relator Ministro José Múcio Monteiro. Brasília: TCU, SEFTI – Secretaria de Fiscalização de Tecnologia da Informação. 2013.
- CASALEA, Giuliano; CHESTAA, Cristina; DEUSSENA, Peter; NITTO, Elisabetta Di; GOUVASA, Panagiotis; KOUSSOURISA, Sotiris; STANKOVSKIA, Vlado; SYMEONIDISA, Andreas; VLASSIOUA, Vlassis; ZAFEIROPOULOSA, Anastasios; ZHAO, Zhiming. **Current and Future Challenges of Software Engineering for Services and Applications.** CF2016, 18-20 October 2016, Madrid, Spain.
- CHERMACK, Thomas J. **Studying scenario planning: Theory, research suggestions, and hypotheses.** Technological Forecasting & Social Change 72 (2005) 59–73.

Elsevier. Received 25 July 2003; received in revised form 12 October 2003; accepted 7 November 2003.

- CRNKOVIC, Ivica. **Component-based Software Engineering – New Challenges in Software Development**. Malardalen University, Department of Computer Engineering, Västerås, Sweden. 2001.
- DERNIAME, J.C.; KABA, B.A.; WASTELL, D. **The Software Process: Modelling and Technology**. Software Process, LNCS 1500, pp. 1-13. Springer-Verlag Berlin Heidelberg 1999.
- FINKELSTEIN, Anthony; KRAMER, Jeff. **Software Engineering: A roadmap. The Future of Software Engineering 2000: 22nd International Conference on Software Engineering**. (pp. pp. 3-24). ACM Press: New York.
- FIRAT, Ayse Kaya; WOON, Wei Lee; Madnick, Stuart. **Technological Forecasting – A Review**. Working Paper CISL# 2008-15, pp. 1–19.
- FITZGERALD, B. **Software Crisis 2.0**. in *Computer*, vol. 45, no. 4, pp. 89-91, April 2012. doi: 10.1109/MC.2012.147.
- GERHARDT, Tatiana Engel; SILVEIRA, Denise Tolfo. **Métodos de Pesquisa**. Universidade Federal do Rio Grande do Sul. 2009.
- GIOVINAZZO, R. A. FISCHMANN, A. A. **Delphi Eletrônico – Uma Experiência de Utilização da Metodologia de Pesquisa e seu Potencial de Abrangência Regional**. Trabalho apresentado no XIV Congresso Latinoamericano de Estrategia. 17, 18 e 19 de maio de 2001. Buenos Aires, Argentina.
- HEIJDEN, Kess van der. **Planejamento por Cenários**. Artmed Editora S.A. São Paulo. 2009
- HICKEY, Ann M. DAVIS Alan M. **A Unified Model of Requirements Elicitation**. Journal of Management Information Systems / Spring 2004, Vol. 20, No. 4, pp. 65–84. 2004. M.E. Sharpe, Inc.
- JUNIOR, Antônio Lourenço; OLIVEIRA, Luiz Cláudio Vieira de; KILIMNIK, Zélia Miranda. **O planejamento de cenários como aprendizado**. Future Studies Research Journal. ISSN 2175-5825 São Paulo, v. 2, n. 1, pp. 03 - 32, Jan./jun. 2010.

- JÚNIOR, Rivalino Matias. **Envelhecimento de software utilizando ensaios de vida acelerados quantitativos**. Tese de doutorado. Universidade Federal de Santa Catarina. Florianópolis, Agosto de 2006.
- KAHN, H.; WIENER, A.J. **The Year 2000: A Framework for Speculation on the Next Thirty-Three Years**. The Macmillan, New York, 1967.
- KAYO, E.K.; SECURATO, J.R., **Método Delphi: Fundamentos, Críticas e Vieses**. Caderno de Pesquisas em Administração, São Paulo, v.1, n.4, p. 51-61, 1º Sem/1997.
- KROEGER, Trent A.; DAVIDSON, Neil J.; COOK Stephen C. **Understanding the characteristics of quality for software engineering processes: A Grounded Theory investigation**. Defence and Systems Institute, University of South Australia, Mawson Lakes, South Australia 5095, Australia. 2013.
- KUOSA, Tuomo. **Towards the dynamic paradigm of futures research: how to grasp a complex futures problem with multiple phases and multiple methods**. Turku School of Economics, Series A-8:2009. PhD Thesis, 2009.
- LEE, Ling-Chu; LEE, Yi-Yang; LIAW, Yi-Ching. **Bibliometric analysis for development of research strategies in agricultural technology: the case of Taiwan**. Received: 21 October 2011 / Published online: 31 August 2012. Akadémiai Kiadó, Budapest, Hungary, 2012.
- MADNI, Hussain Ahmad; ANWAR, Zahid; SHAH, Munam Ali. **Data Mining Techniques and Applications – A Decade Review**. Conference Paper. September 2017. DOI: 10.23919/IConAC.2017.8082090
- MARTINO, Joseph P. **Technological Forecasting for Decision Making**. Elsevier Science Ltd; 2nd edition. January 1983.
- MILLS, Harlan D., "**Management of Software Engineering, The - Part I: Principles of Software Engineering**". The Harlan D. Mills Collection. 1999.
- NUSEIBEH Bashar, EASTERBROOK, Steve. **Requirements Engineering: A Roadmap**. ICSE '00 Proceedings of the Conference on The Future of Software Engineering. Pages 35-46 Limerick, Ireland — June 04 - 11, 2000.

- OSTERWEIL, Leon J. **A Future for Software Engineering?** FOSE '07 2007 Future of Software Engineering, Pages 1-11. IEEE Computer Society Washington, DC, USA. 2007.
- PAETSCH, Frauke; EBERLEIN, Armin; MAURER, Frank. **Requirements Engineering and Agile Software Development.** Agile Surface Engineering (ASE) group at the University of Calgary, Canadá. Proc. WETICE 2003, IEEE.
- PRESSMAN, Roger S.; MAXIM, Bruce R. **Software engineering: a practitioner's approach.** Published by McGraw-Hill Education. 2015.
- PUCINI, Lucas Rebelo Silva; GIFFONI, Mara Gonçalves Pinto; SILVA, Leoni Ferreira da; UTAGAWA, Claudia Yamada. **Comparativo entre as bases de dados PubMed, SciELO e Google Acadêmico com o foco na temática Educação Médica.** Cadernos UniFOA, Volta Redonda, n. 28, p. 75-82, ago. 2015.
- RAMOLLARI, Ervin; DRANIDIS, Dimitris; SIMONS, Anthony J. H. **A Survey of Service Oriented Development Methodologies.** Proc. Second European Young Researchers Workshop on Service Oriented Computing, 2007.
- RANDELL, Brian. Dagstuhl-Seminário 9635: **History of software engineering.** Schloss Dagstuhl, 26-30 de agosto de 1996.
- RAMESH, BALASUBRAMANIAM; PRIES-HEJE, JAN; BASKERVILLE, RICHARD. **Internet Software Engineering: A Different Class of Processes.** Annals of Software Engineering 14, 169–195, 2002. Kluwer Academic Publishers. Manufactured in The Netherlands.
- ROWEA, Gene; WRIGHTB, George. **The Delphi technique as a forecasting tool: issues and analysis.** International Journal of Forecasting 15 (1999)
- RUHE, Günther. **Software Engineering Decision Support – A New Paradigm for Learning Software Organizations.** S. Henninger and F. Maurer (Eds.): LSO 2002, LNCS 2640, pp. 104–113, 2003.
- RUS, Ioana; LINDVALL, Mikael. **Knowledge Management in Software Engineering.** Fraunhofer Center for Experimental. Software Engineering, Maryland. IEEE SOFTWARE May/June 2002.

- SANTOS, Marcio de Miranda; COELHO, Gilda Massari; SANTOS, Dalci Maria dos; FILHO, Lélío Fellows. **Prospecção de tecnologias de futuro: métodos, técnicas e abordagens**. Parcerias estratégicas - número 19 - dezembro/2004. CGEE.
- SCHENATTO, Fernando José Avancini; POLACINSKI, Édio; ABREU, Aline França de; ABREU, Pedro Felipe de. **Critical analysis of future studies: an approach based on history and concepts of the topic**. Gest. Prod., São Carlos, v. 18, n. 4, p. 739-754, 2011.
- SLOCUM, Nikki. **Participatory Methods Toolkit. A practitioner's manual. Method: Delphi**. King Baudouin Foundation and the Flemish Institute for Science and Technology Assessment (viWTA). ISBN 90-5130-506-0. 2005.
- TICHY, Walter F; HABERMANN, Nico; PRECHELT, Lutz. **First Dagstuhl Seminar on Future Directions in Software Engineering**. February 17 - 21, 1992, Schlo Dagstuhl, Germany.
- WAGNER, Stefan; RUH, Melanie. **A Systematic Review of Productivity Factors in Software Development**. arXiv:1801.06475v1 [cs.SE] 19 Jan 2018.
- WANG, Yingxu; KING, Graham. **Software engineering processes: principles and applications**. CRC Press. Taylor & Francis Group. 2000.
- WOHLIN, Claes; AURUM, Aybüke. **Towards a decision-making structure for selecting a research design in empirical software engineering**. Springer Science+Business Media New York 2014.
- WRIGHT, J.; GIOVINAZZO, R. **Delphi: uma ferramenta de apoio ao planejamento prospectivo**, Caderno de Pesquisas em Administração, v. 1, pp. 54–65. 2000.

APÊNDICE A - QUESTIONÁRIO EXPLORATÓRIO

1ª RODADA

Pesquisa acadêmica sobre **O Futuro dos Métodos de Desenvolvimento de Software** utilizando a técnica Delphi de Prospecção do Futuro.

Este trabalho faz parte de pesquisa de Mestrado Acadêmico em Ciência da Computação da Universidade Federal de Pernambuco, utilizando a técnica para Prospecção do Futuro Delphi, do Método de Opinião de Especialistas. Agradeço sua colaboração.

Por favor, responda com base em sua experiência pessoal e conhecimento em desenvolvimento de software, incluindo área de atuação e anos de experiência em engenharia de software, e em fábrica de software, desenvolvimento colaborativo, entre outros. Esta pesquisa estará disponível até o dia 31/05.

- Nome
 - Área de atuação
 - Gênero
 - Tem experiência em Engenharia de Software
1. Considerando que a internet é a principal plataforma para desenvolvimento de software e para o acesso a dados e sistemas, os processos de engenharia de software atuais serão adequados para o software que será desenvolvido no futuro?
 2. Se os processos de engenharia de software atuais não forem adequados para o software que será desenvolvido no futuro, como deverá ser a engenharia de software dentro desse cenário?
 3. O desenvolvimento colaborativo de software está em crescimento. Como será o processo de desenvolvimento do software se o desenvolvimento colaborativo tiver um crescimento muito acentuado?
 4. Se o desenvolvimento colaborativo tiver um crescimento muito acentuado, o que será necessário a partir da evolução das empresas e das aplicações futuras?
 5. O software está cada vez mais especializado e independente na tomada de decisões. Imaginando que o software possa ser considerado um cidadão, como aconteceu com

Sophia, como poderemos desenvolver software com a participação de um "software cidadão"?

6. Quais os atributos de qualidade que são considerados atualmente e quais serão os atributos de qualidade relevantes no futuro?
7. Quais as tendências atuais no desenvolvimento de software?
8. Quais as tendências futuras no desenvolvimento de software?

APÊNDICE B - QUESTIONÁRIO DE AVALIAÇÃO 2ª RODADA

2ª Rodada da Pesquisa acadêmica sobre **O Futuro dos Métodos de Desenvolvimento de Software** utilizando a técnica Delphi de Prospecção do Futuro.

Este trabalho faz parte de pesquisa de Mestrado Acadêmico em Ciência da Computação da Universidade Federal de Pernambuco, utilizando a técnica para Prospecção do Futuro Delphi, do Método de Opinião de Especialistas. Agradeço sua colaboração.

Por favor, responda com base no resumo das respostas dadas pelos especialistas para cada questão, da seguinte forma:

- Concorda totalmente
- Concorda parcialmente
- Discorda parcialmente
- Discorda totalmente
- Não sei/não quero opinar
- Outro: (alguma explicação ou comentário adicional)

1. Considerando que a internet é a principal plataforma para desenvolvimento de software e para o acesso a dados e sistemas, os processos de engenharia de software atuais serão adequados para o software que será desenvolvido no futuro?

Respostas: Sendo a internet a principal plataforma para desenvolvimento de software e para o acesso a dados e sistemas, os processos de engenharia de software atuais não serão adequados para o software que será desenvolvido no futuro, havendo necessidade de adaptação nas metodologias, o uso de ferramentas de mais alto nível, além do que os processos precisam ser constantemente revisados.

2. Se os processos de engenharia de software atuais não forem adequados para o software que será desenvolvido no futuro, como deverá ser a engenharia de software dentro desse cenário?

Respostas: Os processos de engenharia de software atuais são adequados para o software que será desenvolvido no futuro, devendo, no entanto ser observadas as seguintes questões: ser baseado em software como serviços (SaaS), ser feito por

pessoas altamente qualificadas para atender as exigências, a construção do software ser baseada em construção ágil, em modalidade incremental, sem necessidade de prévio desenho completo, Flexível e abrangente para adaptação, cada vez mais baseada no usuário em entregas de valor, simplicidade e com suporte de ferramentas para desenvolvimento distribuído.

3. O desenvolvimento colaborativo de software está em crescimento. Como será o processo de desenvolvimento do software se o desenvolvimento colaborativo tiver um crescimento muito acentuado?

Respostas: as metodologias atuais já contemplam, mas deve permitir que equipes geograficamente distantes possam avançar nos seus projetos, ser voltado para interfaces, necessitará de melhoria na integração e testes, de modo incremental e maior necessidade de ferramentas para dar suporte à gestão e comunicação entre as pessoas.

4. Se o desenvolvimento colaborativo tiver um crescimento muito acentuado, o que será necessário a partir da evolução das empresas e das aplicações futuras?

Respostas: Transparência nas atividades para possibilitar um melhor controle, com boas práticas de comunicação, valorizar as ferramentas de colaboração e acesso remoto aos artefatos desenvolvidos pela equipe, melhoria dos requisitos, e fortalecer o gerenciamento do desenvolvimento.

5. O software está cada vez mais especializado e independente na tomada de decisões. Imaginando que o software possa ser considerado um cidadão, como aconteceu com Sophia, como poderemos desenvolver software com a participação de um "software cidadão"?

Respostas: “Estabelecer claramente a responsabilidades para os integrantes do time, teria a capacidade de minerar fenômenos da sociedade através das redes sociais, tais como reclamações, passeatas, reivindicações, desastres, etc.; sendo assim, a aplicação de técnicas de social mining e machine learning serão imprescindíveis, utilizar o conhecimento agregado destes softwares, dedicação a padronização, dependerá das conquistas no campo da Inteligência Artificial, em especial a eficiência dos algoritmos Learning Machine”.

6. Quais os atributos de qualidade que são considerados atualmente e quais serão os atributos de qualidade relevantes no futuro?

Respostas: “Alta disponibilidade; segurança; velocidade em experimentação, inovação e mudanças. Atuais: segurança, rastreabilidade, maturidade, adequação, conformidade, funcionalidade, confiabilidade, usabilidade, eficiência, manutenibilidade, portabilidade e integração. Futuros: disponibilidade, escalonamento, tolerância a falhas, funcionalidade, confiabilidade, usabilidade, eficiência, aprendizado e auto ajuste, segurança, e performance. A recomendação de atributos mínimos de qualidade não está sendo empregada com a devida responsabilidade. Em especial, se observarmos o cumprimento da Engenharia de Requisitos vimos, por exemplo, que vários sistemas que rodam sobre a Internet não conseguem sequer atender ao requisito da Escalabilidade, o que dirá da segurança então? No futuro a preocupação deveria ser quanto ao real cumprimento da pauta proposta hoje pela Engenharia de requisitos. Não é necessário criar mais indicadores ou métricas de qualidade. Fazer cumprir o que tem. Vamos refletir: softwares estão causando filas no comércio, nos bancos etc. Os modelos de construção estão dissociados da realidade que emerge das demandas sociais”.

7. Quais as tendências atuais no desenvolvimento de software?

Respostas: “Maior modularidade e interação entre diferentes sistemas/serviços, software como serviço, app e integração, desenvolvimento ágil, experimentalismo, agregação incremental, construção de interfaces mais naturais e interativas”.

8. Quais as tendências futuras no desenvolvimento de software?

Respostas: “Conseguir acelerar entregas com valor realmente agregado reduzindo o tempo entre uma identificação de mudança/desenvolvimento e esta característica ir para o ambiente em produção o mais rápido possível, com o nível adequado de qualidade. Software baseado na Internet das Coisas em sistemas de tempo real com sensores. App, integração e interface com o usuário do futuro, Design Driven Development, Inteligência artificial embutida”.

APÊNDICE C - Listar Padrões e Tendências

JCIT.

Probabilidade de Desenvolvimento	1 Estabelecendo claramente responsabilidades para cada integrante / técnicas de social mining e machine learning serão imprescindíveis dedicando-o ao que for padronizável, dependerá das conquistas no campo da Inteligência Artificial em especial a eficiência dos algoritmos de Learning Machine.	2 Modularidade e interação entre diferentes sistemas/serviços. Software como serviço / App e integração / desenvolvimento ágil / experimentalismo / agregação incremental / construção de interfaces mais naturais e interativas.
	3 Adaptações nas metodologias, sistemas de tempo real, streaming IoT, Ferramentas Case, novos paradigmas, métodos ágeis.	4 Voltada para interfaces / melhoria na integração e testes / modo incremental / ferramentas para dar suporte à gestão e comunicação / transparência nas atividades / boas práticas / melhoria na comunicação / ferramentas de colaboração / acesso remoto / pessoal qualificado em interfaces, fortalecimento no gerenciamento, melhoria nos requisitos.
		5 SaaS, pessoas altamente qualificadas, construção ágil, modalidade incremental / Flexível / abrangente / baseada no usuário / desenvolvimento distribuído / Atuais: alta disponibilidade, segurança, velocidade em experimentação / inovação, mudanças, rastreabilidade, maturidade, adequação, conformidade, funcionalidade, confiabilidade / Usabilidade, eficiência, manutenibilidade, portabilidade, integração / Os atributos mínimos de qualidade não estão sendo empregados com a devida responsabilidade. / Futuros: disponibilidade, escalonamento, tolerância a falhas, funcionalidade, confiabilidade, usabilidade, eficiência / aprendizado, auto ajuste, segurança, usabilidade, performance e o real cumprimento da pauta proposta hoje pela Engenharia de requisitos.

P. = Padrão
T = Tendência

Adequação = Auto - Ajuste = Flexível.

o curso -
Foco no
Métricas
Adaptação dos
processos de dev.
P/ não ma
Interi
Cloud