



Universidade Federal de Pernambuco

Centro de Informática

Bacharelado em Sistemas de Informação

RECOMENDAÇÕES PARA MELHORIA DOS PROCESSOS DA METODOLOGIA PAI BASEADAS EM PRÁTICAS ÁGEIS

Elson Rodrigues dos Santos

Recife, 2017

Universidade Federal de Pernambuco

Centro de Informática

Bacharelado em Sistemas de Informação

Elson Rodrigues dos Santos

RECOMENDAÇÕES PARA MELHORIA DOS PROCESSOS DA METODOLOGIA PAI BASEADAS EM PRÁTICAS ÁGEIS

Trabalho de Graduação apresentado ao Curso de Graduação de Sistemas de Informação do Centro de Informática (CIn) da Universidade Federal de Pernambuco (UFPE), como requisito para obtenção do grau de Bacharel em Sistemas de Informação.

Orientador: Hermano Perrelli de Moura

Folha de Aprovação

RECOMENDAÇÕES PARA MELHORIA DOS PROCESSOS DA METODOLOGIA PAI BASEADAS EM PRÁTICAS ÁGEIS

Trabalho de Graduação apresentado ao Curso de Graduação de Sistemas de Informação do Centro de Informática (CIn) da Universidade Federal de Pernambuco (UFPE), como requisito para obtenção do grau de Bacharel em Sistemas de Informação sob orientação do professor Hermano Perrelli de Moura.

Data de Aprovação: / /

Banca Examinadora:

Professor Hermano Perrelli de Moura

Professor Alexandre Marcos Lins de Vasconcelos

Agradecimentos

Agradeço primeiramente a Deus, pelo dom da vida, e por estar sempre me guiando, nos momentos fáceis e difíceis.

Agradeço também a minha família, meu pai Amaro José dos Santos, minha mãe Edmilda Rodrigues dos Santos, por sempre me incentivar a nunca desistir de estudar, pois conhecimento nunca é demais. Ao meu irmão Elvis Rodrigues dos Santos, por me mostrar o quão incrível é ser aluno da Universidade Federal de Pernambuco.

Agradeço aos meus amigos da faculdade, aos amigos do 'busão' e aos meus colegas de turma, em especial Dayse Ferreira, Annemberg Gomes, Mariana Melo e Victor Ferraz, pelo longo trajeto e todos os momentos de estudo e lazer que tivemos juntos.

Agradeço aos meus amigos 'gringos', Greg, Allissa, Bronx, Tenzan e Abigaël, pelas conversas sobre o futuro e os momentos de descontração que tivemos enquanto jogávamos online.

Agradeço a professora Carla Taciana Lima Lourenço Silva, pelo excelente trabalho como professora e também como coordenadora do curso de Sistemas de Informação, por sempre estar disposta a ajudar e tirar dúvidas dos alunos.

Agradeço a meu orientador Hermano Perrelli de Moura, pela confiança, paciência e bom humor ao me guiar nessa tarefa de concretizar o resultado dos anos de estudo.

Agradeço a Paulo Marques, José Danilo e Isadora Oliveira, demais orientandos do professor Hermano, pela cumplicidade durante todo o trajeto para confecção deste trabalho.

Agradeço a professora Cristina Raposo, Lucas Gallindo, Daniel Lago, Liliana Vieira, Christina Nunes e Alba Valéria pela disponibilidade de me receber e pela atenção dada durante as conversas sobre o PAI. Sem a ajuda de vocês este trabalho não teria sido possível.

Aos demais com quem tive a oportunidade de compartilhar vivências, saberes e valores, muito obrigado por tudo. A todos, muito obrigado!

Resumo

O Plano de Ação Institucional (PAI) da UFPE foi desenvolvido com o objetivo de integrar o planejamento entre as unidades organizacionais da universidade, propondo que as unidades acadêmicas administrativas se alinhem para a elaboração de suas propostas anuais. Por se tratar do planejamento operacional, possui uma série de atividades, como formulação de objetivos, cronogramas e avaliação de riscos. Cada atividade é representada por processos, e no ambiente organizacional, nem todo processo ocorre dentro do planejado. Eficiência e agilidade são fatores fundamentais para que os processos atinjam os resultados determinados em seus escopos. Os Métodos Ágeis surgiram como meio de garantir que os processos sejam executados da melhor forma possível, de acordo com suas especificações. Neste trabalho, os processos que fazem parte do PAI serão analisados em busca de alguma deficiência. Aos processos diagnosticados como deficientes, serão propostas recomendações de práticas ágeis com o intuito de melhoria dos mesmos.

Palavras Chave: Plano de Ação Institucional, métodos ágeis, processos, melhoria de processos.

Abstract

The UFPE's Institutional Action Plan (PAI) was developed with the objective of integrating the planning of the university's organizational units, proposing that the administrative academic units align themselves for the preparation of their annual proposals. Because it is operational planning, it has a series of activities, such as the formulation of objectives, schedules and risk assessment. Each activity is represented by processes, and in the organizational environment, not every process occurs within the planned. Efficiency and agility are key factors for processes to achieve the results determined in their scopes. Agile Methods have emerged as a mean of ensuring that processes are performed in the best way possible, according to their specifications. In this work, the processes that are part of the PAI will be analyzed in search of some deficiency. Within the processes diagnosed as deficient, agile practices recommendations will be proposed, with the purpose of improving them.

Keywords: Institutional Action Plan, agile methods, processes, process improvement.

LISTA DE FIGURAS

Figura 1 – Esquema de um processo no PMBOK	26
Figura 2 – Hierarquia dos processos	28
Figura 3 – Ciclo PDCA	30
Figura 4 – Quantidade de ações planejadas por UP's PAI-UFPE-2016	33
Figura 5 – Modelo Tradicional (cascata) e Ágil.....	35
Figura 6 – Processo do Scrum.....	38
Figura 7 – Práticas do Scrum	39
Figura 8 – Processo FDD	43
Figura 9 – Comparação entre TDD e desenvolvimento tradicional	45
Figura 10 – Fluxo do (macro)processo de Planejamento PAI-UFPE-2016	49
Figura 11 – Fluxo do (macro)processo de Monitoramento PAI-UFPE-2016.....	50
Figura 12 – Fluxo do processo de ajuste do planejamento PAI-UFPE-2016.....	51

LISTA DE QUADROS

Quadro 1 – PDCA na gestão de processos.....	30
Quadro 2 – Eixos Temáticos PAI 2016	32
Quadro 3 – Princípios do Manifesto Ágil.....	36
Quadro 4 – Lista de Entrevistados	52

Lista de Abreviações e Siglas

DAP – Diretoria de Avaliação e Planejamento

FDD – Feature-Driven Development

PAI – Plano de Ação Institucional

PDI – Plano de Desenvolvimento Institucional

PMBOK – Project Management Book of Knowledge

PROACAD – Pró-Reitoria de Assuntos Acadêmicos

PROAES – Pró-Reitoria de Assuntos Estudantis

PROEXC – Pró-Reitoria de Extensão e Cultura

PROGEST – Pró-Reitoria de Gestão Administrativa

PROPLAN – Pró-Reitoria de Planejamento, Orçamento e Finanças

SDLC – Systems Development Life Cycle

SIGA – Sistema Integrado de Gestão Acadêmica

TDD – Test-Driven Development

UFPE – Universidade Federal de Pernambuco

Sumário

1	Introdução.....	21
1.1	Contexto.....	21
1.2	Objetivos.....	22
1.2.1	Objetivo Geral	22
1.2.2	Objetivos Específicos	22
1.3	Metodologia.....	23
1.4	Estrutura do Trabalho.....	24
2	Referencial Teórico	25
2.1	Processos Organizacionais	25
2.2	Ciclo PDCA	28
2.3	Plano de Ação Institucional	31
2.4	Métodos Ágeis.....	34
2.4.1	Scrum	38
2.4.2	LSD (Lean Software Development)	41
2.4.3	FDD (Feature-Driven Development)	43
2.4.4	TDD (Test-Driven Development).....	44
2.4.5	XP (eXtreme Programming)	45
3	Análise e Diagnóstico dos Processos.....	48
3.1	Análise dos Processos	48
3.2	Diagnóstico dos Processos	51
3.2.1	Estruturação das Entrevistas	52
3.2.2	Identificação dos Problemas	53
4	Recomendações de Melhoria.....	59
4.1	Problemas Seleccionados.....	59
4.2	Melhorias Recomendadas	59
4.2.1	Falta de Monitoramento no Cadastramento das Ações	59

4.2.2	Integração dos Sistemas que compõem o PAI	60
4.2.3	Escopo Fechado	61
4.2.4	Falta de Automação nos Processos	61
4.2.5	Falta de um Backlog	62
4.2.6	Falta de Objetividade do Sistema.....	62
5	Conclusão e Trabalhos Futuros	64
	Referências	66
	Apêndice	70

1 Introdução

Este capítulo tem como objetivo apresentar o contexto no qual se baseia este trabalho, os objetivos pretendidos com sua execução, a metodologia empregada para que os objetivos sejam alcançados e finalmente é descrita a estrutura deste documento.

1.1 Contexto

O Plano de Ação Institucional - PAI propõe a integração do planejamento entre as unidades organizacionais visando abranger toda a UFPE. A divisão do planejamento vem sendo baseado nos eixos temáticos desde 2012, que são agrupamentos de temas que auxiliam na orientação e no planejamento do trabalho, suscitando questões relacionadas a um determinado assunto e o articulando com outros assuntos (UFPE, 2014, Manual de Elaboração do PAI).

Possui uma visão de planejamento matricial, já que as unidades acadêmicas administrativas precisam de um alinhamento para a elaboração de suas propostas anuais. Também apresenta um dinamismo que permite que seja aprimorado anualmente. O PAI foi desenvolvido tendo como uma das bases o ciclo PDCA.

O PDCA, do inglês *plan-do-check-act*, é uma metodologia de gestão iterativa que consiste na execução de um ciclo, onde processos e produtos são constante e continuamente melhorados. Segundo Vieira Filho (2014, p. 24), “esse método é largamente utilizado na busca da melhoria contínua tão necessária para o sucesso dos negócios”.

No âmbito da melhoria contínua, surgiram métodos e técnicas cujo objetivo é promover otimização e eficiência através de agilidade, que vieram a se tornar o que conhecemos como *Agile* (ou metodologia ágil).

Metodologia Ágil é o termo dado a um conjunto de abordagens de planejamento incremental e iterativo utilizado no desenvolvimento de software, muitas vezes sendo referido como Desenvolvimento Ágil.

Tendo como objetivo inicial a otimização dos processos em desenvolvimento de software, essa prática veio resolver os problemas que

surgiam à medida que a indústria de software evoluía. Entretanto, à medida que a indústria evoluiu, assim também foi com o método ágil. Hoje, as técnicas e filosofias destas metodologias são aplicadas em empresas e organizações de diversas outras áreas, originando vários frameworks, como Extreme Programming (XP), Scrum, Lean Development, Feature-Driven Development (FDD), Kanban, entre outros.

1.2 Objetivos

1.2.1 Objetivo Geral

Este trabalho tem como objetivo geral desenvolver um conjunto de recomendações baseadas em metodologias ágeis visando à melhoria dos processos que fazem parte do Plano de Ação Institucional (PAI), já que os mesmos seguem o modelo cascata, o que faz com que o tempo de resposta desses processos fique comprometido devido às demandas da universidade. Além disso, tais práticas também tem o intuito de tentar corrigir eventuais falhas.

1.2.2 Objetivos Específicos

Para que o objetivo geral seja atingido, as atividades necessárias para tal foram segmentadas em atividades menores, que compõem os objetivos específicos a seguir:

1. Será feita uma análise dos processos que compõem o PAI e, a partir dessa análise, serão diagnosticados aqueles que apresentam algum grau de deficiência;
2. Uma vez que os processos tenham sido devidamente analisados e diagnosticados, serão definidas quais metodologias ágeis se aplicam melhor a cada contexto;
3. Durante a avaliação desses casos, serão propostos conjuntos de práticas ágeis que possam ser aplicadas buscando otimização ou correção da deficiência identificada.

1.3 Metodologia

Para obtenção das informações necessárias ao desenvolvimento deste trabalho, foram adotadas duas técnicas de pesquisa: levantamento dos dados e entrevista.

No levantamento dos dados, foi feita uma pesquisa bibliográfica do material pré-existente, que consistiu em obter todo e qualquer documento referente ao Plano de Ação Institucional, desde que de livre acesso. Esses documentos foram utilizados para identificação dos processos que compõem o plano. Uma vez identificados os principais processos, seguiu-se para a etapa da entrevista.

“A entrevista é uma das técnicas de coleta de dados considerada como sendo uma forma racional de conduta do pesquisador, previamente estabelecida, para dirigir com eficácia um conteúdo sistemático de conhecimentos, de maneira mais completa possível, com o mínimo de esforço de tempo.” ROSA; ARNOLDI (2006) p17.

Gil (1999) admite que a entrevista é seguramente a mais flexível de todas as técnicas de coleta de dados de que dispõe as ciências sociais. Ribeiro (2008) aponta como vantagens da utilização da técnica da entrevista, a flexibilidade na aplicação, a facilidade de adaptação de protocolo (o modo de condução da entrevista pode sofrer modificações), viabilizar a comprovação e esclarecimento de respostas, a taxa de resposta elevada, entre muitas outras.

O tipo de entrevista utilizado para obtenção da segunda parte das informações necessárias seguiu o método semiestruturado. Nesse método, o entrevistador segue um roteiro estabelecido previamente, entretanto, diferindo do método estruturado, onde não é permitido alterar o roteiro para se adaptar à situação, aqui há a liberdade para alterar o roteiro de acordo com o que a situação necessitar, explorando toda e qualquer possibilidade.

Com a realização das entrevistas, buscou-se obter informações referentes a execução dos processos do PAI, assim como dos sistemas que suportam o mesmo em sua totalidade, mas com enfoque em como eles atendem ao propósito do cadastramento e monitoramento das ações institucionais. A eficiência do sistema é um fator crítico, uma vez que a identificação dos processos deficientes e dos passíveis de melhoria depende das informações coletadas com as entrevistas.

Em seguida, uma vez que os processos tenham sido diagnosticados, será feita uma análise de contexto, na qual será definida a melhor abordagem ágil a ser aplicada visando a melhoria. Aqui, serão definidos conjuntos de boas práticas ágeis para os processos identificados como deficientes.

1.4 Estrutura do Trabalho

Esta seção apresenta um guia de como o documento deste trabalho está estruturado:

- **Capítulo 1:** Introdução – Neste capítulo é apresentada uma visão resumida do trabalho, através da contextualização dos temas e do problema abordado, dos objetivos pretendidos bem como a metodologia utilizada.
- **Capítulo 2:** Referencial Teórico – Neste capítulo são abordados os temas utilizados como base para o trabalho, através da análise de literatura sobre os mesmos.
- **Capítulo 3:** Análise e Diagnóstico dos Processos – Neste capítulo são coletadas e analisadas informações referentes aos processos que fazem parte do Plano de Ação Institucional, além da identificação daqueles cujas atividades sejam propensas de melhorias.
- **Capítulo 4:** Proposta de Melhoria – Neste capítulo são sugeridas melhorias para os processos baseadas em práticas ágeis.
- **Capítulo 5:** Conclusão e Trabalhos Futuros – Neste capítulo é apresentada a conclusão do trabalho. Também serão apresentadas possíveis aplicações futuras e algumas limitações que ocorreram ao decorrer do mesmo.

2 Referencial Teórico

Este capítulo apresenta uma visão geral dos temas que embasam o desenvolvimento deste trabalho.

2.1 Processos Organizacionais

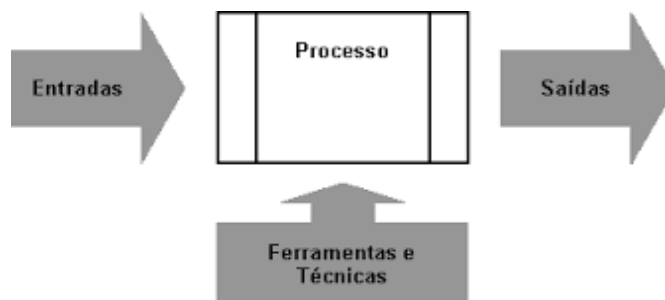
Segundo Harrington (1991), os processos utilizam os recursos da organização para oferecer resultados objetivos aos seus clientes. Para Davenport (1994), seria uma ordenação específica das atividades de trabalho no tempo e no espaço, com um começo, um fim, inputs e outputs claramente identificados, enfim, uma estrutura para ação.

De uma maneira mais formal, processos são conjuntos de atividades agrupadas em determinada ordem a fim de apresentarem início e fim, além de possuírem entradas e saídas. Tais atividades têm o objetivo de gerar resultados plausíveis para o meio em que se executam.

Ainda segundo Harrington (1991), essa ideia de processo como um fluxo de trabalho, com *inputs* e *outputs* claramente definidos e tarefas discretas que seguem uma sequência e que dependem umas das outras numa sucessão clara, vem da tradição da engenharia (que também deu origem à ideia de reengenharia). Os inputs podem ser materiais, como equipamentos e outros bens tangíveis, mas também podem ser informações e conhecimento. Nessa visão, os processos também têm início e final bem determinados. Essa abordagem, característica dos adeptos do aperfeiçoamento de processos acompanhou o raciocínio da engenharia industrial.

A Figura 1 representa, de forma simplificada, o esquema de um processo.

Figura 1 - Esquema de um processo no PMBOK



Fonte: www.devmedia.com.br

Os processos organizacionais podem ser divididos em 3 tipos: gerenciais, de apoio e de negócio ou cliente (Gonçalves, 2000).

- Processos gerenciais – São ligados aos gerentes e à estratégia da organização. Estão diretamente relacionados com a formulação de políticas e diretrizes para o estabelecimento de metas e objetivos. Também incluem a criação e desenvolvimento de métricas de ajustes de desempenho da organização (Garvin, 1998) e as ações que os gerentes realizam para dar suporte aos demais processos de negócio.
- Processos de apoio – também conhecidos como processos de integração organizacional, são centralizados na organização e viabilizam o funcionamento coordenado dos vários subsistemas da organização, garantindo o suporte adequado aos processos de negócio. Estão diretamente relacionados à gestão de recursos necessários ao desenvolvimento dos processos da organização. Esse tipo de processo geralmente produz resultados imperceptíveis para clientes externos, mas que são essenciais para uma gestão efetiva do negócio.
- Processos de negócio – Caracterizam as ações da organização e são suportados por outros processos internos. Segundo Dreyfuss (1996), os processos de negócio são ligados a essência da organização. A soma desses esforços resulta no produto ou serviço recebido pelo cliente externo.

Os processos gerenciais e de apoio são processos de informação e decisão, e podem ser verticais ou horizontais. Os processos classificados como

verticais costumam se referir ao planejamento e orçamento, relacionando-se com alocação de recursos escassos, como fundos e talentos. Já os processos classificados como horizontais são baseados no fluxo de trabalho. Segundo Galbraith (1995), o trabalho nos processos horizontais pode ser realizado de diversas maneiras, gerando três tipos de processos horizontais (laterais): voluntários, formais e coordenados.

- Voluntários – ocorrem por meio do contato voluntário entre os membros do grupo por iniciativa dos envolvidos.
- Formais – definidos previamente por meio de documentos formais.
- Coordenados – exigem times de organização mais complexa e formal.

Galbraith (1995) ainda salienta que os processos horizontais (informação e decisão) são criados para coordenação das atividades que se espalham por várias unidades organizacionais.

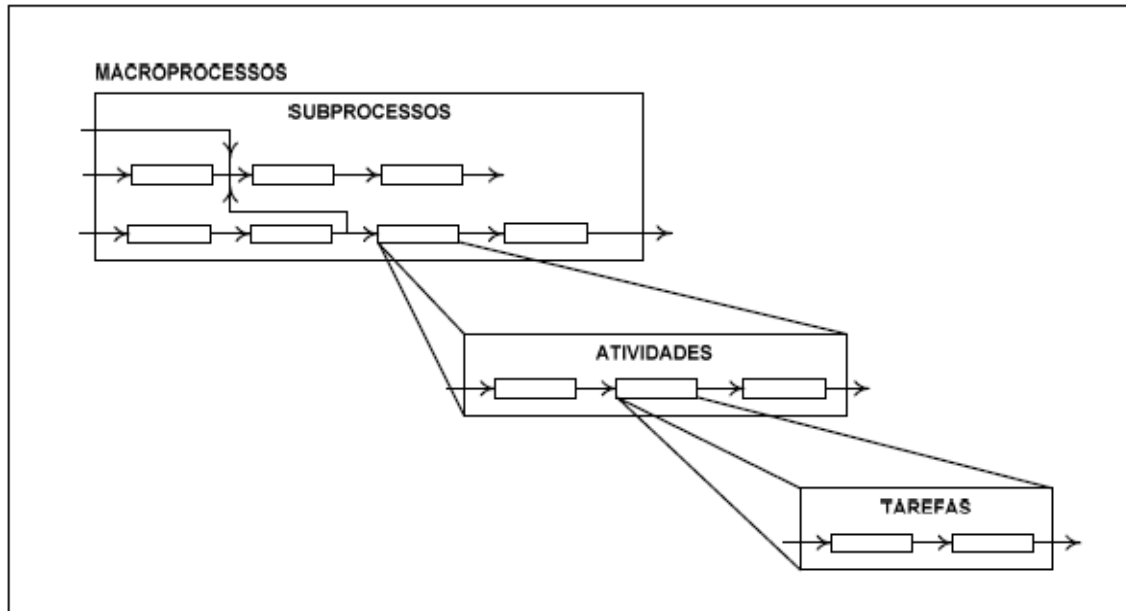
Harrington (1991) apresenta que os processos organizacionais também possuem uma hierarquia, que serve para determinar a prioridade na execução dos processos e a importância dos mesmos.

- Macroprocesso: É um processo geralmente envolvido em mais de uma parte da organização. Sua execução tem impacto significativo no modo como a organização funciona.
- Processo: É um conjunto de atividades gerenciais (conectadas), relacionadas e lógicas que tomam um *input*, acrescentam valor ao mesmo e produzem um *output* para o cliente.
- Subprocesso: É um conjunto de atividades de média e alta complexidade (distintas e interligadas), realizando um objetivo específico em apoio a um processo.
- Atividade: São coisas que ocorrem dentro do processo ou subprocesso. Em geral são desempenhadas por uma unidade (que pode ser uma pessoa ou departamento) para produzir um resultado particular. Fazem parte da maioria dos fluxogramas.

- Tarefa: Conjunto de trabalho a ser executado, envolvendo rotinas, esforço, prazos, etc. Pode ser um único elemento ou subconjunto de uma atividade.

A Figura 2 representa esquematicamente essa hierarquia.

Figura 2 – Hierarquia dos processos



Fonte: HARRINGTON (1993, p.34)

2.2 Ciclo PDCA

O ciclo PDCA ou simplesmente PDCA, do inglês *Plan-Do-Check-Act*, é um modelo de gestão de processos que se baseia no princípio de melhoria contínua. É aplicado largamente ao redor do mundo por inúmeros tipos de organizações. Surgiu nos anos 20, mas só passou a ser amplamente conhecido na década de 50, quando William Edwards Deming o apresentou em diversas palestras pelo Japão. Até esse momento, era conhecido como ciclo de Shewhart ou ciclo de Deming.

Segundo Marshall Junior et al (2006), o PDCA é um gerenciamento que promove melhoria contínua e reflete a todo momento a busca pela mesma. Ele diz:

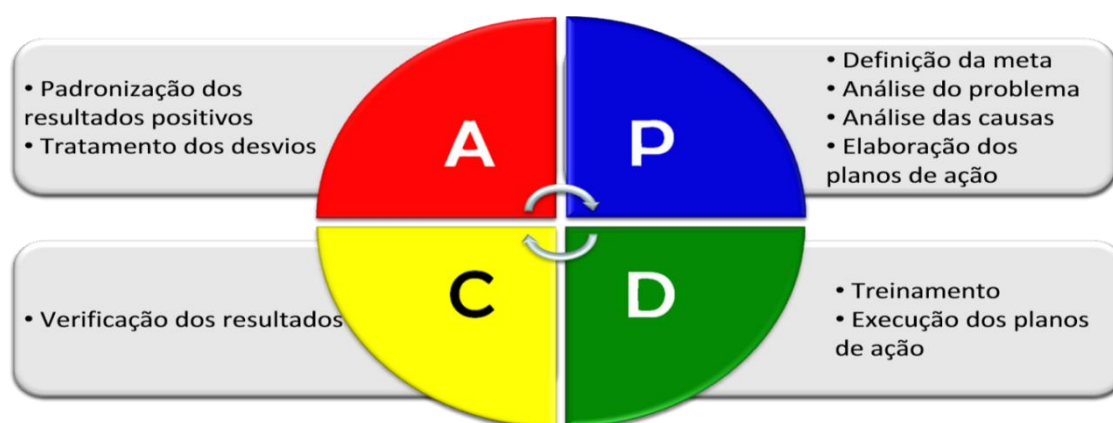
O ciclo PDCA é um método gerencial para a promoção da melhoria contínua e reflete, em suas quatro fases, a base da filosofia do melhoramento contínuo.

O objetivo do PDCA é trazer otimização através de melhoria, percorrendo as etapas que formam seu ciclo.

Cada letra constitui uma etapa do processo, e devem ser seguidas em ordem. Na etapa P (*Plan*) - Planejamento, são estabelecidos planos/diretrizes, de acordo com os objetivos da empresa, além do plano de ação. Na etapa D (*Do*) - Execução, todos os planos definidos na etapa de Planejamento devem ser postos em prática. Aqui também são coletadas informações para o mapeamento das próximas etapas. Na etapa C (*Check*) - Verificação, é feita uma análise comparativa entre as etapas anteriores, checando se as ações realizadas na etapa de Execução atendem às demandas definidas na etapa de planejamento, se há conformidade entre as partes. Segundo Vieira Filho (2014, p. 25), esta é uma etapa puramente gerencial, que verifica se o que foi executado está de acordo com as metas estabelecidas. Na etapa anterior, são coletados dados das ações e estes dados são analisados nesta etapa e comparados com o planejado. Na etapa A (*Act*) – Agir, são realizadas as ações corretivas embasadas em todas as informações produzidas nas etapas anteriores. Uma vez que as correções sejam aplicadas, deve-se repetir o processo, começando novamente o planejamento, daí o ciclo de melhoria contínua.

A Figura 3 esquematiza essa descrição feita anteriormente.

Figura 3 - Ciclo PDCA



Fonte: julianakolb.wordpress.com

Do ponto de vista de processos, Alencar e Souza (2013) definem metas para cada etapa do ciclo PDCA, no âmbito da gestão de processos e resultados a serem alcançados. O Quadro 1 apresenta essas metas.

Quadro 1 - PDCA na gestão de processos

	Plan	Do	Check	Act
O quê?	Planejar a gestão por processos	Modelar e otimizar os processos	Implantar e checar os resultados	Atuar sobre o processo: correção ou melhoria
Como?	Transformar metas do Planejamento Estratégico Institucional em planos de ação. Estabelecer diretrizes e especificações. Formar equipes e priorizar atividades.	Modelagem AS IS (desenho de como o processo é realmente executado, com erros e acertos) e TO BE (desenho da situação ideal). Realizar simulações com alternativas de soluções. Propor a melhoria do processo.	Desenvolver os processos com sugestões de melhorias Implantar novos processos. Monitorá-los e controlá-los. Realizar estatísticas com base em indicadores.	Fazer análise comparativa da situação anterior e da atual. Efetuar ajustes necessários. Monitorar continuamente o processo para identificar oportunidades de novas melhorias.

Fonte: Manual de Gestão por Processos – Ministério Público Federal (2013). Adaptado pelo autor

Ainda segundo Alencar e Souza (2013), aplicando o PDCA se consegue:

- Avaliar desempenho;
- Analisar comparativamente o planejado e o realizado;
- Analisar desvios;
- Tomar ações corretivas;
- Acompanhar a eficiência das ações implementadas;
- Captar informações que auxiliem a tomada de decisão.

2.3 Plano de Ação Institucional

Formular objetivos, prever as atividades, programar o tempo e avaliar os riscos são apenas algumas das tarefas envolvidas na elaboração de um planejamento operacional (UFPE, 2014, Manual de Elaboração do PAI). O PAI foi criado em 2011 pela Pró-Reitoria de Planejamento, Orçamento e Finanças, a PROPLAN. A Diretoria de Avaliação Institucional e Planejamento (DAP) também trabalhou na elaboração, estruturação e execução deste plano.

As estratégias adotadas pela UFPE no processo de elaboração do seu planejamento estão alicerçadas em três níveis: estratégico (o Plano Estratégico Institucional PEI-2013/2027), tático (o Plano de Desenvolvimento Institucional-PDI-2014/2018) e operacional. O instrumento de planejamento de curto prazo (em nível operacional) é o Plano de Ação Institucional - PAI (UFPE, 2016, Relatório do PAI).

O PAI propõe a integração do planejamento entre as unidades organizacionais visando abranger toda a UFPE. A divisão do planejamento vem sendo baseada desde 2012 em eixos temáticos. O Quadro 2 apresenta os eixos temáticos do PAI.

Quadro 2 - Eixos Temáticos PAI 2016

01.	Formação Acadêmica
02.	Internacionalização
03.	Pesquisa, Inovação, Extensão e Cultura
04.	Meio Ambiente, Sustentabilidade e Acessibilidade
05.	Gestão
06.	Pessoas, Saúde e Qualidade de Vida
07.	Vida Estudantil
08.	Infraestrutura e Segurança
09.	Informação e Comunicação

Fonte: Relatório do PAI 2016

Esses eixos temáticos não são estáticos, estando sempre sujeitos ao melhoramento que também abrange o próprio PAI e vem sendo trabalhado a cada ano, conforme abordagens do PDCA, que faz parte da base de seu planejamento.

O PAI possui uma visão de planejamento matricial, já que as unidades acadêmicas administrativas precisam de um alinhamento para a elaboração de suas propostas anuais. Essas unidades referidas como “Unidades de Planejamento (UP’s)”, englobam todas as pró-reitorias, todas as 12 diretorias dos Centros Acadêmicos e todos os diretores dos “órgãos suplementares” (UFPE, 2016, Relatório do PAI).

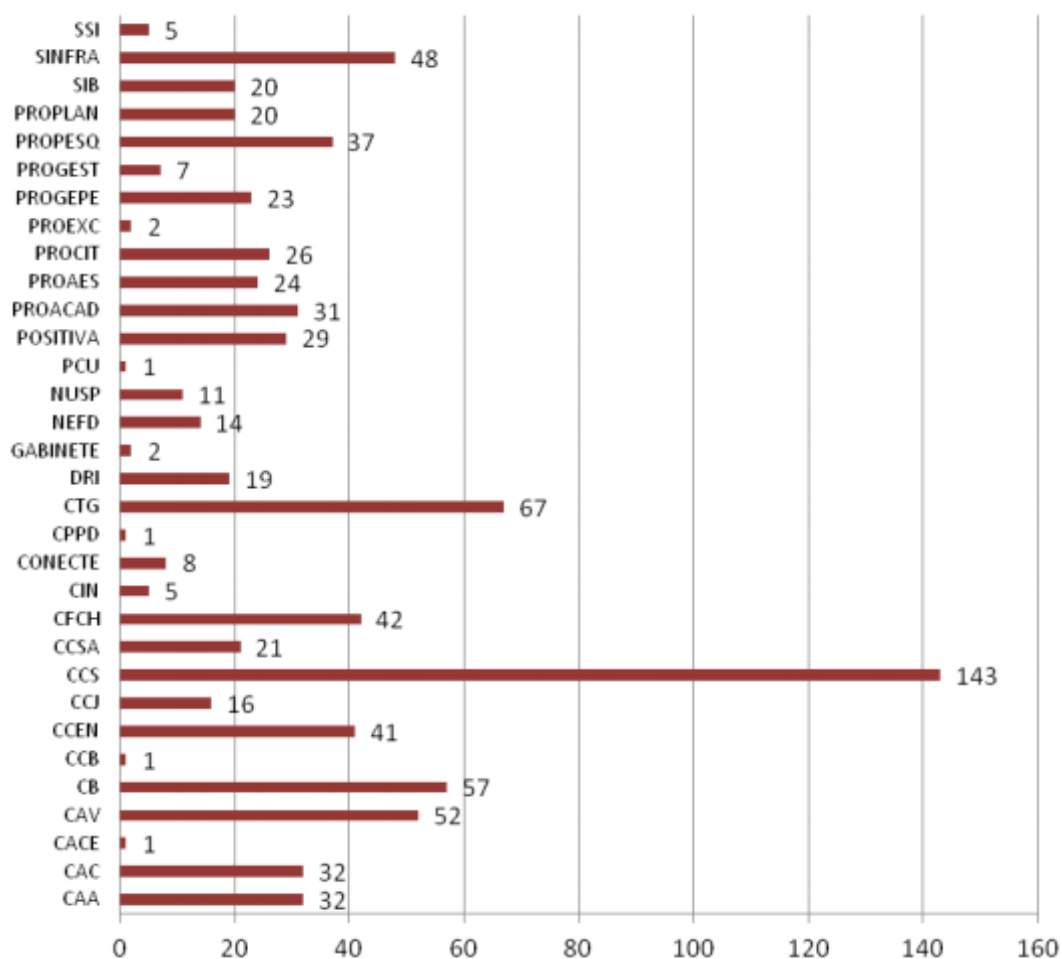
Segundo Nascimento (2016), através do PAI, os gestores têm um panorama claro e preciso dos recursos disponíveis, quais os objetos a serem alcançados, como e onde os recursos serão empregados, quais os resultados esperados, quais avanços foram efetuados ao longo do exercício administrativo, e quais mudanças são necessárias para atingir integralmente os objetivos. Sua proposta principal é integrar o planejamento das unidades organizacionais de modo que abranja toda a universidade.

No ano de 2016 o planejamento foi feito usando a Plataforma SIGAPLAN, disponível no sistema acadêmico da UFPE, o sistema SIGA, e o monitoramento foi feito usando o sistema *redmine* (UFPE, 2016, Relatório do PAI). Antes desta data, o acompanhamento era feito pela “tecnologia WEP”,

que se refere aos programas do pacote Office da Microsoft, o Word, Excel e PowerPoint (ALMEIDA, 2016).

Segundo Nascimento (2016), Ação é o conceito do que precisa ser feito para a instituição atingir suas metas. Para cada ação ou conjunto delas, existe um gerente de ação. O gerente é o responsável por criar, monitorar e avaliar a ação. Ele também é o responsável financeiro, incluindo prestar contas pela ação. Os gerentes de ações de um determinado eixo temático se reportam à figura de um Coordenador. Este Coordenador tem o dever, dentre outros, de acompanhar a execução das ações que ele coordena e cadastrar os gerentes de ações. Ele também pode ser gerente e coordenador simultaneamente. A Figura 4 apresenta a quantidade de ações planejadas por UP's referentes ao ano de 2016.

Figura 4 – Quantidade de ações planejadas por UP's PAI-UFPE-2016



Fonte: Relatório do PAI 2016

A partir do calendário do ciclo de vida do PAI, trimestralmente as unidades organizacionais se reúnem para realizar o diagnóstico e monitoramento do PAI. Nestas reuniões as unidades organizacionais prestam contas da execução das ações propostas, apresentando, sobretudo o percentual de recursos financeiros empregados na data de cada avaliação, tendo como base o montante total provisionado (Nascimento, 2016). Todas essas informações são compiladas e utilizadas no desenvolvimento do Manual de Elaboração do PAI, um documento produzido anualmente que contém o rol de Eixos Temáticos abordados, a descrição de cada eixo bem como os objetivos a serem alcançados, dados orçamentários, e outras informações que subsidiarão na elaboração do plano (Nascimento, 2016). Este manual irá nortear a execução do PAI no ano seguinte.

2.4 Métodos Ágeis

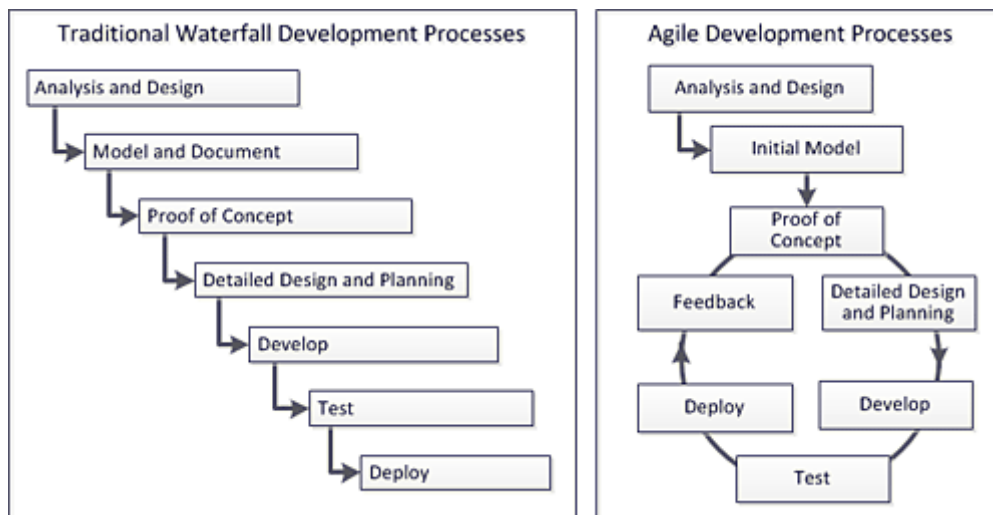
Desde o surgimento da indústria de *software*, técnicas para o desenvolvimento de soluções vêm sendo criadas e evoluídas ao longo dos anos. O modelo cascata, que foi um dos pontos de partida, determinava que projetos só poderiam ir para frente à medida que suas etapas fossem completadas em sua totalidade. Entretanto, segundo Dr. Winston Royce, ao apresentar o periódico “Managing the Development of Large Software Systems” em 1970, softwares não deveriam ser desenvolvidos como um automóvel ou uma linha de produção, onde cada parte é integrada em fases sequenciais, já que para que uma fase seja iniciada, é preciso que a anterior tenha sido devidamente concluída. Seu discurso gerou uma grande discussão na indústria de software, que até então seguia uma abordagem dirigida a planos.

A partir de então, novos métodos começaram a ser desenvolvidos buscando ultrapassar essa barreira imposta com a evolução na complexidade dos projetos. Os Métodos Ágeis surgiram em resposta à baixa *performance* das abordagens anteriores. O termo ‘ágil’ se refere ao desenvolvimento acelerado, buscando manter o mesmo nível de produção, ou até mesmo superando-o. Métodos Ágeis referenciam um conjunto de abordagens de planejamento

incremental e iterativo utilizado no desenvolvimento de software, podendo também ser referido como Desenvolvimento Ágil.

Segundo Beck (2004), o modelo ágil acredita que cada projeto precisa ser tratado de forma diferente e os métodos existentes precisam ser adaptados para melhor atender aos requisitos do projeto. Highsmith (2009) salienta que quando se trata de desenvolvimento ágil, em vez de seguir o padrão SDLC num único grande processo, o ciclo de desenvolvimento se dividiria em partes menores, chamadas iterações, e cada uma dessas iterações englobaria as etapas do desenvolvimento tradicional. A Figura 5 apresenta o modelo tradicional de desenvolvimento e o modelo ágil.

Figura 5 - Modelo Tradicional (cascata) e Ágil



Fonte: msdn.microsoft.com/

A partir daí, muitos modelos ágeis foram desenvolvidos, cada um com uma abordagem diferente, voltada a um determinado tipo de projeto, mas ainda classificáveis dentro do modelo ágil de desenvolvimento. Segundo Cohn & Ford (2003), abordagem ágil se relaciona à abordagem “*inspect & adapt*” da engenharia, onde ciclos e sessões de *feedback* são curtas.

Em 2001, dezessete desenvolvedores se reuniram para discutir sobre as técnicas que haviam desenvolvido, e esse encontro acabou por gerar o Manifesto Ágil.

O Manifesto Ágil é um documento voltado para o desenvolvimento de software, contendo valores e princípios que norteiam desenvolvedores na

produção de atividades relacionadas ao software ágil. Este manifesto indica que as abordagens ágeis são sobre a entrega de bons produtos aos clientes, operando em um ambiente adaptável e com uma comunicação bem orientada. Eles tentam fornecer um compromisso entre nenhum processo e muitos processos e entre nenhuma documentação e muita documentação.

Os valores do manifesto consistem em:

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

O Quadro 3 apresenta os 12 princípios que fazem parte do Manifesto Ágil.

Quadro 3 - Princípios do Manifesto Ágil

Sr.No.	Principle
1.	Highest priority is to satisfy the customer through early and continuous delivery of valuable software.
2.	Welcome changing requirements ,even late in development.
3.	Deliver working software frequently,from a couple of weeks to a couple of months.
4.	The most efficient and effective method of conveying information to and within a development team is face to face conversation.
5.	Business people and developers must work together daily throughout the project.
6.	Continuous attention to technical excellence and good design enhances teams.
7.	Simplicity--the art of maximizing the amount of work not done is essential.
8.	Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
9.	At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.
10.	Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.
11.	Working software is the primary measure of progress.
12.	The best architectures, requirements, and designs emerge from self-organizing teams.

Fonte: A Comparative study of Agile Software Development Methodology and traditional waterfall model (2015)

Cada princípio se relaciona diretamente com os valores do manifesto, de modo que é possível fazer um mapeamento entre eles. Por exemplo, o valor representado pela ‘colaboração com o cliente’ se relaciona com o 4º e 8º princípios. Já o valor ‘indivíduos e operações’ se relaciona com o 5º e 6º princípios. Há uma definição para métodos ágeis que de certa forma integra e

resume os valores e princípios do manifesto, que segundo Scott W. Ambler (2005-2012) diz o seguinte:

An iterative and incremental (evolutionary) approach to software development which is performed in a highly collaborative manner by self-organizing teams within an effective governance framework with "just enough" ceremony that produces high quality solutions in a cost effective and timely manner which meets the changing needs of its stakeholders.

Além da criação do Manifesto Ágil, o grupo de desenvolvedores envolvidos também criou um grupo denominado *Agile Alliance*, que vem evoluindo ao longo dos anos e atualmente é uma organização sem fins lucrativos que se empenha em apoiar pessoas que se aventuram no meio ágil, proporcionando uma vasta experiência de aprendizado aos seus membros. Suas atividades incluem conferências, que trazem a comunidade reunida, um site cheio de informações sobre *Agile* e a Comunidade *Agile*, acesso a recursos valiosos criados por membros da comunidade e iniciativas que abordam áreas específicas de interesse na comunidade ágil e fornecem suporte para grupos comunitários locais. A *Agile Alliance* tem como filiais oficiais a *Agile Alliance Brazil* e a *Agile Alliance New Zeland*.

Os Métodos Ágeis, que se referem a qualquer técnica de desenvolvimento de software 'mais rápida que o normal', são tão diversificados que descrever todos eles levaria muito tempo, então apenas aqueles que foram utilizados neste trabalho terão sua descrição apresentada mais adiante. Esses métodos foram Scrum, Lean Development, FDD, TDD e XP (eXtreme Programming).

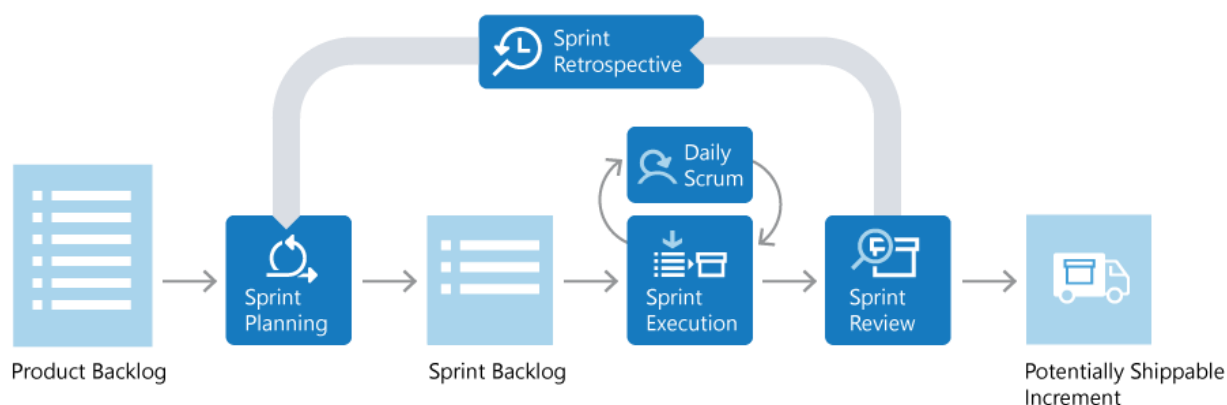
2.4.1 Scrum

O Scrum é tido como umas das melhores técnicas de desenvolvimento ágil. Consiste em processos iterativos e incrementais que servem para todo tipo de trabalho ou produto. Segundo Schwaber e Beedle (2001), é o mais flexível, adaptável, empírico, iterativo e produtivo método para atender às necessidades das indústrias de software.

O Scrum não requer ou provê nenhum método/prática específico de desenvolvimento de software para ser utilizado. Na verdade, ele requer certo grau de práticas de gerenciamento e ferramentas, em diferentes fases, para evitar problemas de imprevisibilidade e complexidade (Rising e Janoff, 2000).

No Scrum, cada ciclo de iteração é denominado *Sprint*, e no fim de cada *Sprint* deve ser entregue ao cliente/usuário algo de valor. O processo de desenvolvimento de cada entrega ocorre num prazo de 30 dias, por isso Scrum é mais utilizado em projetos de curta duração. A Figura 6 apresenta o processo.

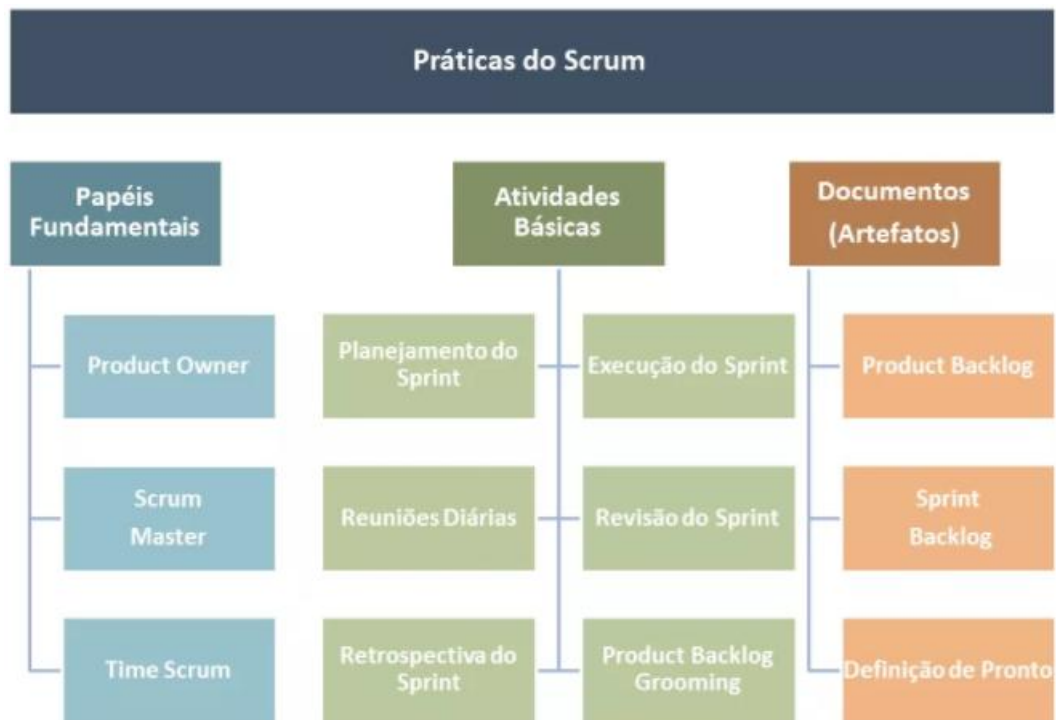
Figura 6 – Processo do Scrum



Fonte: www.visualstudio.com

Como se pode ver, cada etapa do processo corresponde a uma atividade e uma série de artefatos (documentos), que, somados a papéis fundamentais (atribuídos) e mais alguns outros artefatos, compõem a Figura 7. Essa figura representa uma esquematização da prática do Scrum.

Figura 7 – Práticas do Scrum



Fonte: www.mindmaster.com.br

Papéis Fundamentais:

- *Product Owner* – é a pessoa responsável por alocar recursos e definir funcionalidades que farão parte do *Product Backlog*.
- *Scrum Master* – é a pessoa responsável por ajudar a todos os envolvidos a entender os valores, princípios e práticas do Scrum. Ele ajuda o Time Scrum a desenvolver sua própria abordagem respeitando as particularidades da organização, além de ajudar a equipe a resolver problemas e fazer melhorias no uso do Scrum.
- Time Scrum – é a junção de todas as pessoas envolvidas em uma equipe multidisciplinar, e que é responsável pela concepção, construção e testes do produto. A ideia principal é que o Time Scrum se auto-organize para determinar a melhor maneira de realizar o trabalho para atingir a meta estabelecida pelo *Product Owner*. O time tem

tipicamente entre 5 e 9 pessoas e seus membros devem ter coletivamente todas as habilidades necessárias para produzir, com qualidade, software funcionando.

Atividades básicas:

- Planejamento do *Sprint* – Durante o planejamento do Sprint, o Time Scrum e o *Product Owner* devem chegar a um acordo sobre qual o objetivo do Sprint. Com este objetivo, eles determinam quais os itens do *backlog* devem ser priorizados para serem executados neste Sprint.
- Execução do *Sprint* – É o trabalho que o Time Scrum deve realizar para atingir os objetivos definidos durante o planejamento.
- Reuniões Diárias (*Daily Scrum*) – Todos os dias, idealmente no mesmo horário, os membros do Time Scrum devem realizar uma reunião com tempo definido (15 minutos ou menos), chamado *Daily Scrum*, onde é muito comum que o tema abordado seja o andamento e colaboração de cada participante no projeto. Esta reunião também é muitas vezes chamada de *Stand-Up Meeting*, por causa de uma prática recomendada para que a reunião seja feita em pé.
- Revisão do *Sprint* – É uma reunião informal, onde a apresentação mostra o que foi alcançado no Sprint. O objetivo desta atividade é verificar e adaptar o produto que está sendo construído.
- Retrospectiva do *Sprint* – O objetivo é verificar necessidades de adaptações no processo de trabalho. Ocorre depois da Revisão da Sprint e antes da reunião de planejamento da próxima Sprint.
- *Product Backlog Grooming* – São reuniões com o objetivo de aprimorar o *Product Backlog*. *Grooming* se refere à atividade de criar e de refinar os itens do *Product Backlog*, estimando o tamanho e esforço de cada item.

Documentos (Artefatos):

- *Product Backlog* – É uma lista contendo todas as funcionalidades desejadas para um produto, cujo conteúdo é definido pelo *Product Owner*. O *Product Backlog* é um documento que está constantemente evoluindo. Os itens podem ser adicionados, excluídos e revisto pelo *Product Owner* por conta de mudanças nas condições de negócios, ou conforme a compreensão da equipe Scrum sobre o produto aumenta.
- *Sprint Backlog* – é uma lista de tarefas que o Time Scrum se compromete a fazer em um Sprint. Os itens do *Sprint Backlog* são extraídos do *Product Backlog* pela equipe, com base nas prioridades definidas pelo *Product Owner* e a percepção da equipe sobre o tempo que será necessário para completar as várias funcionalidades.
- Definição do Pronto – ‘*Definition of Done*’ é um acordo formal do Time Scrum que define claramente quais são os passos mínimos para a conclusão de um item potencialmente entregável. Serve como um contrato entre o Time Scrum e o *Product Owner*, garantindo que todo o produto gerado pelo projeto estará dentro dos padrões de qualidade estabelecidos.

No geral, Scrum é amplamente aplicado em projetos de todos os tipos, não só no desenvolvimento de software, pelo fato de ser fácil de implantar e oferecer alta adaptabilidade. Ele tem se mostrado bastante eficaz como meio de melhoria de processos.

2.4.2 LSD (Lean Software Development)

Lean Software Development é uma adaptação das práticas de produção *Lean*, que considera o gasto de recursos para qualquer objetivo que vá além da criação de valor para o cliente final como um desperdício, logo um alvo de eliminação, para o domínio do desenvolvimento de software. Vem crescendo com o apoio de uma cultura pró-*lean* dentro da comunidade ágil.

Partindo da perspectiva do cliente que consome um produto ou serviço, "valor" é definido como qualquer ação ou processo que um cliente estaria disposto a pagar. Em sua essência, é uma filosofia orientada à eficiência e eficácia de processos, centrada em criar mais valor com menos trabalho.

LSD deriva do *Lean Manufacturing*, e é um conjunto de princípios, valores e boas práticas que tem como objetivo a redução do desperdício, a melhoria da qualidade e a maximização do valor entregue ao cliente. Mary e Tom Poppendieck escreveram um livro sobre *LSD*, e por causa do envolvimento deles com desenvolvimento ágil de software, seus conceitos passaram a ser amplamente aceitos dentro da comunidade ágil.

Assim como o *Lean Manufacturing*, o *LSD* possui sete princípios para o desenvolvimento de software (Pawar, 2015):

1. Eliminar o desperdício – Tudo que não agrega valor ao cliente é considerado como desperdício.
2. Amplificar conhecimento – Priorizar a comunicação e o *feedback* contínuo entre equipes e usuários durante o processo de desenvolvimento de software.
3. Adiar decisões – Deixar as decisões e comprometimentos para o último momento responsável, permitindo coletar informações e ter experiências para fortalecer a tomada de decisão.
4. Entregas rápidas – Maximizar o retorno sobre investimento do projeto, entregando software de valor de forma rápida e contínua.
5. Fortalecer o time – Criar um ambiente onde a equipe trabalhe de forma auto-organizada e auto-dirigida.
6. Construir qualidade – Garantir qualidade no desenvolvimento do software.
7. Otimizar o todo – Entender que o software concluído é muito mais que a soma das partes entregues e verificar como ele está alinhado com os objetivos da empresa.

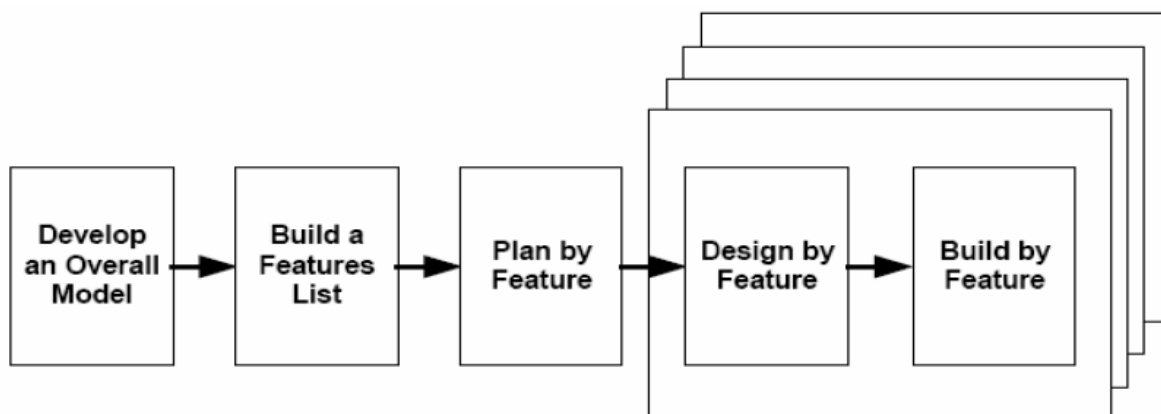
Uma vez que desenvolvimento ágil de software é um termo abrangente para um conjunto de métodos e práticas baseados nos valores e princípios

expressos no Manifesto ágil, o *Lean Software Development* é considerado um método de desenvolvimento de software ágil.

2.4.3 FDD (Feature-Driven Development)

Feature-Driven Development é um método de desenvolvimento que não foca no desenvolvimento como um todo, mas sim nas fases de construção e design. O objetivo principal do FDD é gerar entregas mais frequentes e tangíveis, juntamente com o rastreamento preciso dos relatórios de progresso (Awad, 2005). Se divide em 5 fases, sendo que 3 são feitas no início do projeto, enquanto que as 2 restantes são iterativas e dão suporte ao processo de desenvolvimento ágil com adaptações rápidas para mudanças tardias em requisitos e necessidades comerciais. A Figura 8 representa as fases do processo.

Figura 8 – Processo FDD



Fonte: A Comparison between Agile and Traditional Software Development Methodologies (2005)

Detalhando as fases (Awad, 2005):

- *Develop an overall model* – um passo-a-passo de alto nível do escopo do sistema e seu contexto é produzido pelo *domain expert* para os membros do time e o chefe de arquitetura. Documentação de requisitos como casos de uso são desenvolvidos.

- *Build a features list* – uma lista categorizada das funcionalidades que dão suporte aos requisitos é produzida.
- *Plan by feature* – o time de desenvolvimento pede um conjunto de funcionalidades de acordo com as prioridades e dependências, e o atribui aos programadores chefes. Além disso, as classes identificadas na primeira fase são atribuídas a desenvolvedores individuais (*class owners*).
- *Design by feature + Build by feature* – as funcionalidades são selecionadas a partir dos conjuntos de funcionalidade e as equipes necessárias para desenvolvê-las são escolhidas pelos *class owners*. Ambas as etapas são procedimentos iterativos onde a equipe produz diagramas de sequência para as funcionalidades atribuídas. Estes diagramas são transmitidos aos desenvolvedores que implementam os itens necessários para suportar o design de uma funcionalidade particular. Pode haver várias equipes simultaneamente projetando e criando seu próprio conjunto de funcionalidades. O código desenvolvido é então testado unitariamente e inspecionado. Após uma iteração bem sucedida, as funcionalidades são movidas para o código principal.

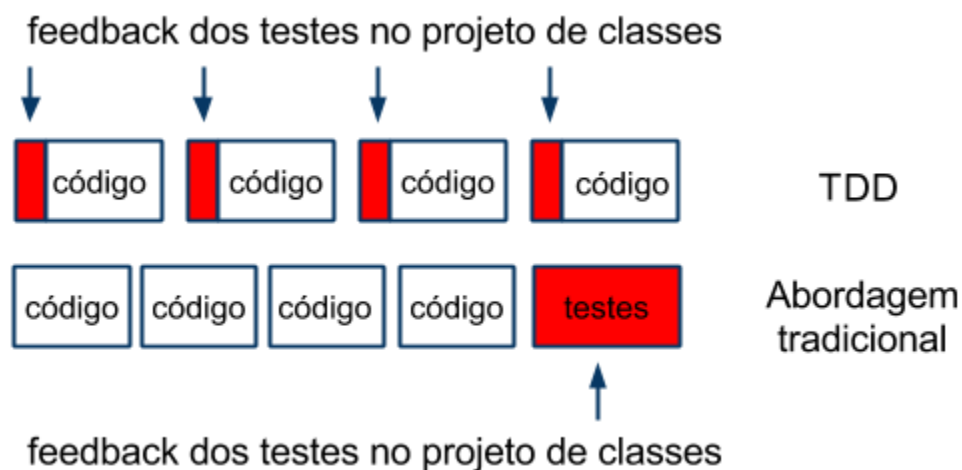
2.4.4 TDD (Test-Driven Development)

Test-Driven Development é um método de desenvolvimento de *software* que depende da repetição de um ciclo de desenvolvimento muito curto: primeiro o desenvolvedor escreve um caso de teste automatizado (supostamente falho), que define um melhoramento desejado ou nova função. Então, produz a menor quantidade de código para passar nesse teste, e finalmente refatora o novo código para padrões aceitáveis. Para Aniche (2014), a prática de TDD agrega muitos benefícios ao processo de desenvolvimento. O primeiro deles, e mais claro, são os benefícios na qualidade externa do produto. Uma das principais vantagens de utilizar TDD é o tempo de resposta nos feedbacks.

Segundo Aniche (2014), o desenvolvedor obtém *feedback* do teste. A diferença é justamente na quantidade de *feedback*. Quando o desenvolvedor

escreve os testes somente ao acabar a implementação do código, ele passou muito tempo sem retorno. Ao praticar TDD, o desenvolvedor divide seu trabalho em pequenas etapas. Ele escreve um pequeno teste, e implementa um pedaço da funcionalidade. E repete. A cada teste escrito, o desenvolvedor ganha *feedback*. Quanto mais cedo o desenvolvedor receber *feedback*, melhor. Quando se tem muito código já escrito, mudanças podem ser trabalhosas e custar caro. Ao contrário, quanto menos código escrito, menor será o custo de mudança. E é justamente isso que acontece com praticantes de TDD: eles recebem *feedback* no momento em que mudar ainda é barato. A Figura 9 exemplifica essa diferença nos *feedbacks* citada anteriormente.

Figura 9 – Comparação entre TDD e desenvolvimento tradicional



Fonte: <http://tdd.caelum.com.br/> - Adaptado pelo autor

2.4.5 XP (eXtreme Programming)

XP (*eXtreme Programming*), segundo Qureshi (2012), é uma das metodologias ágeis mais conhecidas e amplamente utilizadas para desenvolvimento de software. Pode ser caracterizado por ciclos de desenvolvimento curtos, planejamento incremental, *feedback* contínuo, dependência de comunicação e design evolutivo (Beck, 2009).

XP defende “entregas” frequentes em ciclos de desenvolvimento curtos, que visam melhorar a produtividade e introduzir checkpoints onde os clientes podem introduzir novos requisitos. O termo ‘extreme’ se refere ao senso

comum de que os princípios e práticas desse método são levados a níveis extremos.

XP possui 4 valores principais que apoiam a motivação e satisfação da equipe. Esses valores incluem (Madi, Dahalin, & Baharom, 2011):

- Comunicação – Em um projeto XP, a comunicação é bidirecional e com base em um sistema de loops de pequenos feedbacks entre os membros da equipe. O cliente trabalha de perto com os desenvolvedores para explicar e agendar as funcionalidades desejadas. Os desenvolvedores lidam com a perspectiva técnica. Os clientes comunicam sua satisfação com o progresso do produto para a equipe de desenvolvimento.
- *Feedback* – Com *feedback* suficiente, a equipe pode medir o sistema e saber onde eles estão e quão longe o sistema está das funcionalidades necessárias. O *feedback* concreto também permite que o cliente solicite uma mudança e veja esses requisitos ou ajustes implementados dentro de um curto período.
- Coragem – Este é um valor importante e é promovido pelos outros três valores. É necessário em todos os níveis. Com coragem, o participante joga para vencer. Alterar um código em desenvolvimento sem gerar bugs e sem comprometer a velocidade, exige muita coragem e responsabilidade.
- Simplicidade – para atender rapidamente às necessidades do cliente, quase sempre um dos valores mais importantes é simplicidade. Normalmente o que o cliente quer é muito mais simples do que aquilo que os programadores constroem.

As práticas de XP consistem em (Beck, 1999):

- Planejamento - O programador estima o esforço necessário para a implementação das histórias do cliente, e o cliente decide o escopo e o calendário dos lançamentos baseados em estimativas.

- Entregas curtas – Uma aplicação é desenvolvida em uma série de pequenas sessões atualizadas frequentemente. Novas versões são lançadas a qualquer momento, diariamente até mensalmente.
- Metáfora - Envolve a concepção do sistema com base na metáfora compartilhada entre o cliente e os programadores.
- Design simples – Códigos e testes unitários são simples.
- Testes - O XP usa uma abordagem de desenvolvimento orientada por teste para garantir que todos os testes unitários sejam satisfatórios e sejam executados corretamente.
- Refatoração – Torna o código mais fácil de entender ao remover a redundância no mesmo.
- Programação por pares – 2 programadores trabalham no mesmo código simultaneamente.
- Propriedade coletiva – Cada programador é responsável pela melhoria do código sempre que possível.
- Integração contínua – O novo código deve ser ajustado e integrado com o sistema existente.
- 40-horas semanais – O programador deve trabalhar no máximo 40 horas por semana. Mais que isso será considerado como um problema.
- Cliente on-site – Se refere ao envolvimento do cliente durante o ciclo de vida do projeto.
- Padronização de código - Os programadores escrevem o código da mesma maneira de acordo com os padrões concordados.

3 Análise e Diagnóstico dos Processos

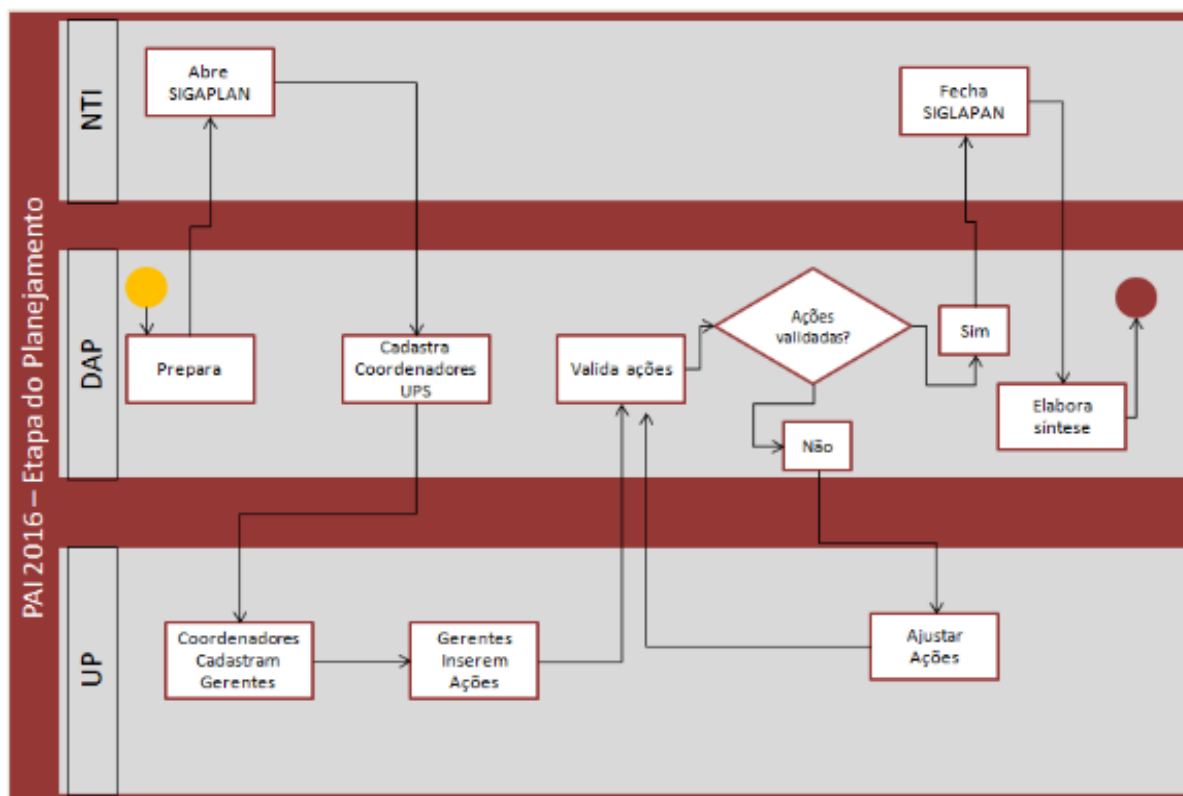
Aqui começamos a analisar diretamente as informações referentes ao PAI. Para realização dessa análise, foram coletadas e analisadas informações provenientes de duas fontes: documentação pré-existente, que continha relatórios, manuais, atas, etc., e através de um levantamento realizado com pessoas que trabalham com o PAI, como gerentes de ação e coordenadores de planejamento de várias UP's espalhadas pela UFPE, além daqueles que trabalharam diretamente com o planejamento e desenvolvimento do mesmo. Nesse levantamento, as informações obtidas se deram por meio de entrevistas.

No processo de preparação para as entrevistas, foi levado em consideração que todos os possíveis entrevistados deveriam ter algum grau de envolvimento com o PAI. Nesse contexto, coordenadores de planejamento e gerentes de ação foram selecionados para tal. O contato com os candidatos foi feito através do envio de e-mails e telefonemas.

3.1 Análise dos Processos

Quando o Plano de Ação Institucional foi criado, seus processos se iniciaram apenas em volta da administração central da universidade e das pró-reitorias, por não ter tido adesão suficiente. Mas com o tempo, eles foram expandidos para abranger também os centros acadêmicos e as demais unidades administrativas da UFPE. Recordando que o PAI teve início em 2012 e vem sendo atualizado e melhorado a cada ano. Dentre os processos principais que compõem o plano, os processos identificados foram os seguintes: *abertura do sistema, desenvolvimento da proposta, validação da proposta, monitoramento (e execução), e fechamento do sistema*. Os processos de abertura, monitoramento e fechamento são considerados macroprocessos nesse meio. Os macroprocessos de abertura e fechamento do sistema estão contemplados na Figura 10.

Figura 10 – Fluxo do (macro)processo de Planejamento PAI-UFPE-2016 (DAP)



Fonte: Fluxo dos processos de planejamento e gerenciamento PAI-UFPE-2016, DAP

É possível observar nesse macroprocesso que o processo de validação está incluso no mesmo como uma tarefa. Entretanto, essa “tarefa” se mostrou muito mais complexa do que sua simples representação.

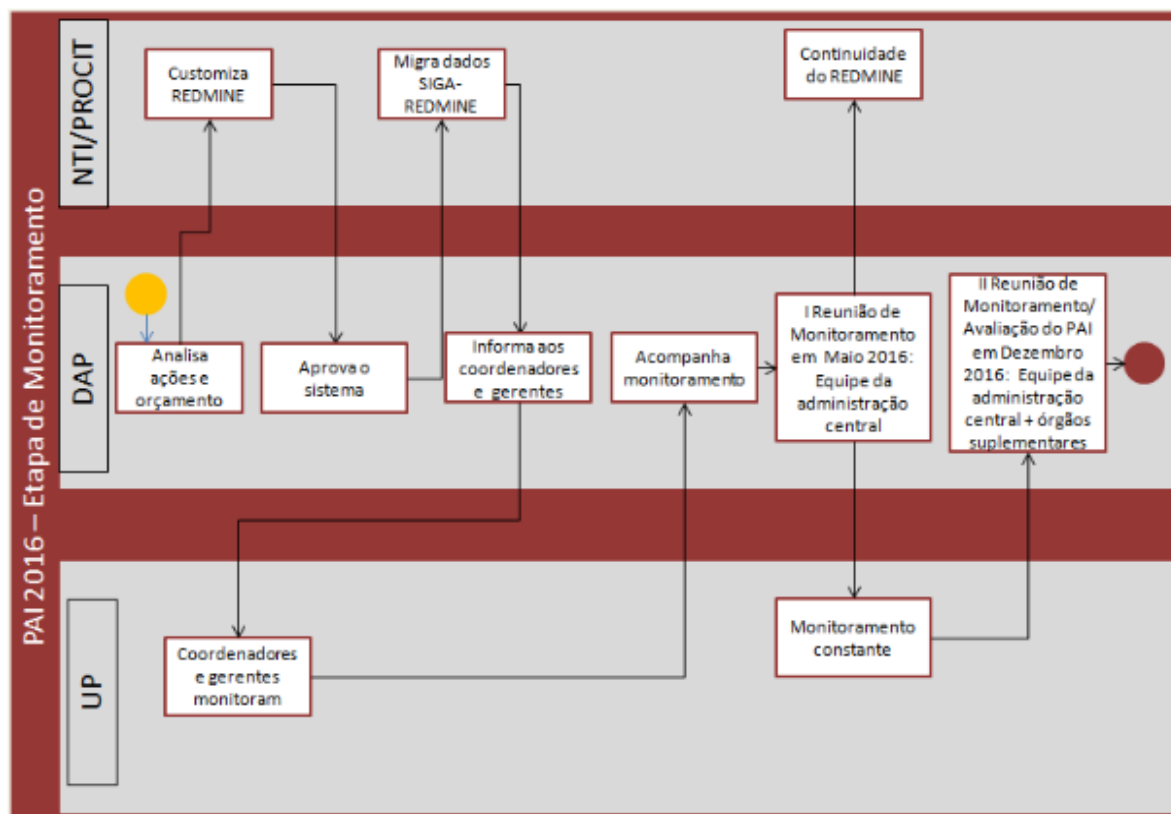
O macroprocesso de monitoramento, representado na Figura 11, contempla as ações que foram cadastradas no sistema e receberam aprovação. Nesse ponto, as ações que foram cadastradas no SIGAPLAN são direcionadas para o servidor *redmine*. É por esse sistema que os gerentes e coordenadores de ação monitoram o status das ações cadastradas e se organizam para prestar contas durante as reuniões de monitoramento.

Vale ressaltar que ações que foram aprovadas não são exatamente realizáveis. Muitas ações podem ser aprovadas durante o processo de validação, que envolve análise orçamentária e alocação de recursos, mas isso não significa que elas serão executadas durante os ciclos (período entre o fechamento e a abertura do sistema para cadastramento de ações). Pelo fato das ações representarem as metas das UP's, elas devem ser cadastradas no

sistema, independente do prazo definido em seus escopos. Uma ação pode ser planejada à longo prazo, ser cadastrada e aprovada, porém ela pode nunca chegar a ser executada.

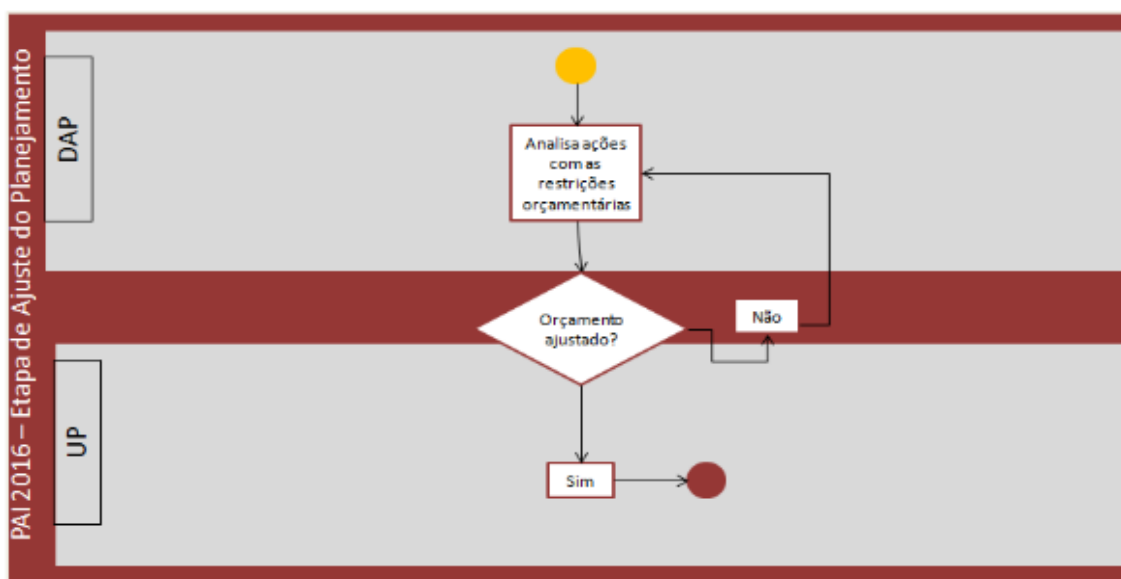
As Figuras 11 e 12 mostram, respectivamente, o macroprocesso de monitoramento e o processo de análise orçamentária.

Figura 11 – Fluxo do (macro)processo de Monitoramento PAI-UFPE-2016 (DAP)



Fonte: Fluxo dos processos de planejamento e gerenciamento PAI-UFPE-2016, DAP

Figura 12 – Fluxo do processo de ajuste do planejamento PAI-UFPE-2016 (DAP)



Fonte: Fluxo dos processos de planejamento e gerenciamento PAI-UFPE-2016, DAP

O processo que representa a análise orçamentária (Figura 12) fica localizado dentro do processo de monitoramento (Figura 11), mais precisamente na tarefa de ‘Analisar ações e orçamento’, onde é na verdade um subprocesso bem mais detalhado. O processo de desenvolvimento da proposta é interno a cada UP, e se refere às ações que são planejadas pelos coordenadores e gerentes, durante o planejamento das metas daquela UP.

Ainda foi possível identificar que os microprocessos que compõem o PAI são todos padronizados, independente da unidade administrativa, sem uma característica (atividade) específica àquela unidade. Nesse caso, tais processos não possuem nenhuma documentação.

Pelo fato dessas informações serem provenientes da documentação referente ao PAI 2016 e a documentação referente ao PAI 2017 ainda estar sendo produzida, é possível que parte das informações já estejam desatualizadas.

3.2 Diagnóstico dos Processos

Tendo identificado os principais processos que definem a essência do PAI, foram realizadas entrevistas cujo objetivo era obter a opinião dos usuários (coordenadores de planejamento e gerentes de ação) sobre a utilização do

sistema e quais as contribuições que o Plano de Ação Institucional trouxe para a universidade. Dentro dessa opinião, buscou-se identificar também possíveis problemas quanto à utilização do sistema.

3.2.1 Estruturação das Entrevistas

O Quadro 4 apresenta o cronograma das entrevistas, com o cargo dos entrevistados e o setor da universidade (no caso, unidade de planejamento) a qual pertencem.

Quadro 4 – Lista de Entrevistados

Código de Identificação	Unidade de Planejamento	Cargo desempenhado
P01	DAP	Gerente de Ação
P02	PROGEST	Gerente de Ação
P03	PROGEST	Gerente de Ação
P04	PROAES (NASE)	Gerente de Ação
P05	PROAES	Coordenador de Planejamento
P06	PROPLAN	Coordenador de Planejamento/Gerente de Ação

Fonte: O Autor

Foi acordado com cada entrevistado que seus nomes não seriam mencionados no trabalho, apenas seu cargo, enquanto usuário, e a unidade de planejamento onde trabalha. Cada um recebeu um código de identificação (P01, P02 e assim por diante) para ajudar no entendimento de quem discutiu o quê.

O roteiro das entrevistas se encontra no apêndice deste trabalho. O período designado para execução das entrevistas foi de 6 a 20 de novembro.

Cada conversa foi realizada na unidade de planejamento onde os entrevistados trabalham, e o contato com eles foi feito através de e-mails para confirmar a disponibilidade e telefonemas para agendar um horário.

O registro das entrevistas foi feito durante a execução das mesmas, onde cada resposta e informação adicional foram transcritas ao decorrer da conversa. Os problemas então identificados foram listados na seção seguinte, assim como uma transcrição da entrevista que embasa o problema em questão.

3.2.2 Identificação dos Problemas

Foi unânime da parte dos entrevistados o quanto o PAI beneficiou a universidade. Como ferramenta de planejamento, pelo fato de tornar 'público' o que está sendo planejado em cada unidade através do sistema onde as ações são cadastradas; por gerar uma cobrança implícita que 'motiva' os gerentes e coordenadores a tentar cumprir o máximo possível do que foi planejado; por permitir a interação com ações de outras unidades, visto que antes cada unidade só interagiu com suas próprias ações; no caso de ações relacionadas a mais de uma UP, facilitar a priorização das ações ao permitir que a administração central tome ciência do que está sendo feito pelas UP's (prestação de contas, justificativa por parte das UP's de por que tal ação não foi concluída dentro do prazo, ver se o planejamento está alinhado com o PDI); e principalmente por automatizar atividades que antes ocorriam de maneira puramente manual.

Individualmente, cada entrevistado apontou problemas relacionados a seu dia-a-dia, no que diz respeito às ações tomadas em relação ao sistema e as atividades que desempenham no mesmo. Todos os problemas identificados durante a discussão foram listados, mas nem todos estão sujeitos à correção através de práticas ágeis, então eles não terão uma proposta de melhoria apresentada no capítulo seguinte.

Os problemas identificados durante as entrevistas, de acordo com a descrição dos coordenadores de planejamento e gerentes de ação participantes, foram os seguintes:

- **Falta de monitoramento no cadastro de ações:** Não existe uma verificação das ações que são cadastradas no sistema, havendo por muitas vezes o recadastro de uma ação. Essas ações repetidas geram volume de dados desnecessário.
 - *“Não existe um registro para cada ação que é cadastrada, então uma mesma ação pode aparecer diversas vezes durante uma consulta no sistema.” – P01.*
 - *“Tem muita ação que aparece repetida se for feita uma busca justamente por não existir um ‘log’ com as informações das ações cadastradas.” – P03.*
- **Integração dos sistemas:** A migração das informações entre o SIGAPLAN e o *redmine* é feita de forma manual (durante o processo de monitoramento), o ‘upload’ das ações é feito pelo usuário manualmente (em alguns casos).
 - *“Faço a atualização das minhas ações direto no banco de dados, tudo na ‘munheca’.” – P01.*
 - *“[...] As ações são lançadas no SIG@ e transferidas no REDMINE, no meu caso não há, aparentemente, integração entre os usuários SIG@ e REDMINE, eu só consigo achar minhas ações procurando manualmente e atualizando uma a uma.” – P03.*
- **Escopo dos processos fechados:** Seguem o modelo cascata e isso prejudica bastante as ações que se modificam ao longo do tempo, pois no fim do processo elas não condizem mais com o planejamento inicial, estando, portanto, desatualizadas.

- *“Muitas das ações que são cadastradas passam por modificações no escopo por causa de orçamento, recursos, tempo, e não existe um meio de atualizar essa mudança no sistema, então quando o relatório de acompanhamento é gerado, aquela ação específica já não existe mais por que as informações cadastradas não batem.”*
– P01.
- **Falta de automação nos processos:** como por exemplo, a migração dos dados entre os sistemas que dão suporte ao PAI.
- **Falta de um *backlog*:** Não existe um *backlog* para as ações que são/estão cadastradas.
 - *“Se existisse um backlog para guardar as informações das ações, não haveria mais o cadastro repetido. Ele também serviria para ajudar no relatório de acompanhamento que é gerado a cada três meses.”* – P01.
- **Ineficiência no suporte ao usuário:** foi constatado que há uma demora no atendimento das requisições dos usuários ou simplesmente os pedidos não são atendidos.
 - *“Encaminhei isso ao departamento responsável em junho e nunca tive sequer resposta.”* – P03 (22/11/17).
 - *“O suporte é muito precário, às vezes situações são comunicadas a PROPLAN para que ela articule com o NTI, só que há uma demora ou nada é resolvido”* – P05.
- **Perda de informação no sistema:** Durante o direcionamento de páginas do sistema com preenchimento de formulários, a informação cadastrada não é carregada no servidor.
 - *“Muitas vezes enquanto preenchia as informações durante o cadastro de uma ação, ao confirmar no fim e realizar uma*

consulta para checar se tinha sido de fato registrada, vi que parte das informações estava faltando, sendo que aquele campo que está vazio foi preenchido durante o cadastro, todas as informações foram devidamente inseridas nos campos.” – P04.

- **Falta de instruções de uso do sistema:** Falta de conhecimento específico sobre a utilização do sistema por parte de coordenadores e gerentes, que podem cometer erros durante a execução de suas atividades no sistema.
 - *“Não existe um guia, um relatório, ou qualquer documentação que ensine aos usuários sobre a utilização e funcionalidades do sistema.” – P03.*
 - *“Não somos orientados sobre como usar o sistema.” – P04.*
 - *“Ninguém nos ensina sobre os processos e atividades do sistema, seja o básico ou o essencial.” – P05.*
- **Falta de avaliação do sistema pelo usuário:** Não existe uma avaliação de satisfação do sistema pelo usuário.
 - *“Às vezes o sistema deixa a desejar e eu não posso reclamar sobre isso. Na verdade mesmo se eu reclamar, ninguém faz nada.” – P05.*
- **Falta de compromisso por parte dos usuários:** Alguns dos coordenadores e gerentes não fornecem todas as informações necessárias durante o cadastro de ações. A intenção dos gestores também não é definida de maneira correta, o que inclui a definição de cortes e a negociação de quais ações devem realmente constar no plano da UP.
 - *“A verificação das ações ainda acontece de forma precária, pois nem todos fornecem as informações necessárias. As informações*

que não chegam tornam difícil um posicionamento em relação à tomada de ações corretivas.” – P06.

- *“As intenções dos gestores não são definidas corretamente, não é feita uma limitação do que deve ou não ser feito, com relação as ações e aos cortes dessas ações.” – P06.*

- **Limitação na atividade dos usuários (coordenador de planejamento):** As atividades do coordenador tornam-se limitadas assim que encerra o prazo de cadastro dos gerentes de ação no sistema. Com o fim desse prazo, o coordenador não pode mais realizar nenhuma modificação nas ações.

- *“Não tenho liberdade para trabalhar as ações e a maioria das atividades só é liberada durante o período de cadastro dos gerentes. No caso, são os gerentes que tem toda liberdade para tratar as ações, enquanto que o coordenador só pode consultar. Fiz um acordo com os gerentes que coordeno para que eu seja aquela que trata das ações. De certo modo, eles são gerentes apenas em nome, pois sou eu que ‘manuseio’ as ações e tudo mais.” – P05.*

- **Sistema fixo:** Não é permitida a realização de mudanças no escopo das ações (edição), as informações cadastradas são fixas (já que não podem ser editadas).

- *“As ações deveriam ser dinâmicas, podendo ser alteradas a qualquer momento, mas depois que você cadastra, é tudo fixo. Tinha que ser trabalhada a flexibilidade do sistema, que só é ‘maleável’ no prazo do cadastro dos gerentes. Quando esse prazo encerra, as atividades ficam todas bloqueadas.” – P05.*

- **Bloqueio de funções no sistema:** Determinadas funções são bloqueadas quando o prazo de cadastro de ações termina, não

permitindo ao coordenador/gerente a modificação de informações quando necessário.

- *“[...] Quando esse prazo encerra, as atividades ficam todas bloqueadas.” – P05.*
- *“[...] e a maioria das atividades só é liberada durante o período de cadastro dos gerentes.” – P05.*
- **Falta de objetividade do sistema:** A interface do sistema não é objetiva o bastante para que os usuários façam exatamente o que precisam. Como não há um ‘manual do usuário’, torna ainda mais complicado a situação dos mesmos na execução das atividades.
 - *“As vezes quando estou preenchendo os campos para cadastrar uma ação, procuro saber com outro gerente como ele fez, e descubro que um campo que preenchi ele deixou em branco e vice-versa. ” – P04.*
 - *“O sistema não é objetivo pra me dizer ‘essa informação vem aqui, enquanto que essa aqui vem nessa outra parte...’. A gente tem que se virar pra entender o quê vai aonde.” – P05.*

4 Recomendações de Melhoria

Baseando-se nos resultados obtidos no capítulo anterior, dada a listagem de problemas relacionados ao sistema do PAI, foi preciso selecionar aqueles cuja proposta de melhoria poderia ser feita através da aplicação de métodos ágeis. Tendo como base os métodos ágeis compilados no capítulo do referencial teórico, foi feita uma análise, buscando aqueles com maior aptidão para servir ao propósito da proposta de melhoria.

4.1 Problemas Selecionados

Nesse contexto, os problemas selecionados foram:

1. Falta de monitoramento no cadastro de ações.
2. Integração dos sistemas que compõem o PAI.
3. Escopo fechado.
4. Falta de automação nos processos.
5. Falta de um backlog.
6. Falta de objetividade do sistema.

4.2 Melhorias Recomendadas

Buscando solucionar tais problemas, soluções baseadas em práticas ágeis foram desenvolvidas. Vale lembrar que os problemas a serem trabalhados foram aqueles considerados críticos, e que a proposta de solução destes pode acabar gerando uma possível solução para os demais problemas que foram identificados.

4.2.1 Falta de Monitoramento no Cadastramento das Ações

Para solucionar esse problema, foi pensado em fazer uso de *sprints*, que é uma prática de Scrum. Seriam realizadas reuniões mensais entre a equipe de coordenadores e gerentes de cada UP para checar o andamento das ações (uma vez que o relatório de acompanhamento só é gerado trimestralmente), bem como os status das mesmas. Se alguma ação se tornar inviável, por qualquer que seja o motivo, não será preciso esperar até o fim do ciclo para

tomar as providências cabíveis. Também poderia ser adotado o *Daily Scrum* (reuniões diárias com a equipe, em cada UP), que seria adaptado em um 'weekly scrum' (reunião semanal).

4.2.2 Integração dos Sistemas que compõem o PAI

Há o problema da integração entre as contas dos usuários para o SIGAPLAN e *redmine*, o que pode ocasionar em perda de informação durante a migração dos dados.

Para solucionar esse problema, foi pensado na aplicação de XP. Primeiramente, as equipes responsáveis pelos sistemas se reuniriam para analisar os códigos dos mesmos. Assim que fosse feita essa análise, seria realizado um planejamento, onde seria verificado qual(is) a(s) chance(s) de integrar os sistemas, de preferência em sua totalidade.

No caso dessa integração completa não poder ser feita, os responsáveis pelos sistemas contatariam os usuários, e de acordo com o *feedback* dos mesmos sobre a utilização do sistema e atividades que fossem mais prioritárias (segundo a prática XP – Game Planning, onde a equipe coleta *stories* dos usuários do sistema), seriam elaboradas estratégias para se chegar a uma solução. Para essas estratégias seriam aplicadas as três últimas etapas do FDD, *plan by feature* e o conjunto *design by feature + build by feature*. Aqui, seriam definidas as funcionalidades dos sistemas que precisariam ser integradas de acordo com o que foi coletado pela equipe. Uma vez que fosse decidida uma solução e começasse a ser implementada, seria aplicado o TDD, onde seriam realizados testes unitários das funcionalidades que seriam integradas, que seriam validados juntamente com os usuários, que participariam dos testes e dariam *feedback*.

No caso de ser possível uma integração completa dos sistemas, para que não houvesse *delay* na execução das atividades, seria introduzido um sistema alternativo, que fosse semelhante aos já utilizados em suas funcionalidades. Todas as atividades seriam transferidas temporariamente para o novo sistema enquanto a integração do SIGAPLAN e *redmine* estivesse em andamento. Assim que a integração fosse concluída, as atividades iriam para o

sistema integrado. Outra alternativa seria fazer a integração por partes, começando pelas atividades mais externas, deixando por fim as atividades principais.

4.2.3 Escopo Fechado

Grande parte dos processos que fazem parte do PAI estão sujeitos ao escopo fechado. Os processos que englobam o cadastro de ações seguem o modelo tradicional cascata, o que significa que é feito um planejamento inicial, antes que as ações comecem de fato a serem executadas, e que só é possível ir adiante se o planejamento for executado de forma plena. Como muitas ações sofrem modificação nos escopos (mudanças no planejamento, alocação de recursos, questões de orçamento, etc.) durante o decorrer do prazo de três meses, não é possível modificá-las no momento que a mudança é pensada.

Através da aplicação do Scrum, dada sua natureza incremental, seria possível alterar o escopo das ações sem comprometer, ou com um comprometimento mínimo, o planejamento que foi realizado à priori durante a definição daquela ação, ou a tempo de adaptá-las a fim de evitar defasagem. Por exemplo, durante a execução das *sprints* para acompanhamento das ações seriam checados os status das mesmas. Se houvesse alguma mudança no escopo, então as modificações necessárias seriam adotadas. Com isso, quando chegasse o tempo de apresentar o relatório de prestação de contas, as ações que sofreram modificações ao longo do período estariam atualizadas no relatório, evitando assim a discrepância das informações, que estariam desatualizadas caso contrário.

4.2.4 Falta de Automação nos Processos

Assim como na integração dos sistemas, seria preciso primeiro analisar o código dos processos que já foram automatizados. Para processos que ainda não saíram do papel, seria preciso analisar a estrutura deles para poder implementá-los.

A solução, através da aplicação de XP, FDD e TDD. Pelo FDD, seriam definidas funcionalidades para os processos (não automatizados) e revisadas as funcionalidades dos que já são automatizados. Do XP faríamos uso do

Game Planning, para coletar dos usuários *feedback* sobre quais as funcionalidades que eles esperam nos processos automatizados. Pelo FDD, essas funcionalidades seriam priorizadas de acordo com a necessidade, e a partir disso a equipe responsável pelo desenvolvimento trataria de cada uma. Do TDD, seria definido um código simples e fácil de testar unitariamente (no módulo de cada funcionalidade) referente à funcionalidade que foi priorizada anteriormente. Esse código será refatorado à medida que obtiver êxito nos testes até que se torne o mais objetivo possível.

A integração contínua (XP) garantiria que a cada novo código testado e aprovado, seja feita a integração do mesmo no sistema até que todos os processos sejam automatizados e estejam funcionais.

4.2.5 Falta de um Backlog

A criação de um backlog, que é uma prática de Scrum, onde estariam registradas informações de todas as ações cadastradas no sistema, onde os coordenadores e gerentes de ação seriam responsáveis pelo monitoramento das mesmas. Esse backlog poderia ser de acesso individual por UP, ou conter uma parte específica para os coordenadores e gerentes. Também serviria para evitar o cadastro de ações repetidas, uma vez que as informações referentes àquela ação já estariam presentes no sistema, e no caso seria permitido ao gerente editar as informações, que posteriormente seriam avaliadas pelo coordenador, que aprovaria ou não essa edição. Serviria também como base para verificação e funcionaria em conjunto com o relatório de acompanhamento gerado a cada três meses.

4.2.6 Falta de Objetividade do Sistema

Alguns usuários disseram possuir certa dificuldade na utilização do sistema, já que não há nenhuma documentação que sirva ao propósito de instrução. Foi constatado que a interface não é objetiva, o que pode gerar descuidos durante a realização das atividades.

Para solucionar isso, aplicaria-se Lean para 'enxugar' a interface, ficando apenas com o essencial, além de oferecer um a descrição dos campos com o que deve ser preenchido em cada um. E Scrum pela transparência, com uma

documentação de linguagem simples que ajudaria o usuário no entendimento das atividades (e dos processos). Essa documentação solucionaria o problema da falta de conhecimento sobre a utilização do sistema.

5 Conclusão e Trabalhos Futuros

A análise dos processos que compõem o Plano de Ação Constitucional na busca por deficiências mostrou que mesmo um processo estruturado de maneira lógica e objetiva pode apresentar falhas. Além do técnico, fatores sociais e humanos também podem contribuir para a incapacidade dos usuários. Apesar deste trabalho propor melhorias baseadas em práticas ágeis, grande parte dos problemas identificados possui solução que independe disso. Questões como mudança nas políticas institucionais podem contribuir para melhoria, e até solução, de muitos dos problemas que foram listados.

A melhoria dos processos também segue esse caminho de fatores técnicos e humanos, mas, além disso, ela é dependente da tecnologia, que está sempre em evolução, e isso faz com que as técnicas utilizadas para o melhoramento também estejam sujeitas a evoluir, já que elas podem se tornar obsoletas e não conseguir dar conta das demandas emergentes.

As propostas de melhoria definidas ao longo deste trabalho são nada mais que sugestões, que foram feitas levando em consideração o fator tecnológico dos processos, os métodos ágeis e por fim os fatores humanos dos usuários. Tais propostas podem sim vir a ser adotadas, pois não há uma solução específica para cada caso, e elas servirão como base para planejamentos futuros. Entretanto, não foi possível apresentar os resultados obtidos juntamente com as práticas propostas para os *stakeholders* com o intuito de obter uma avaliação para validá-las. Tanto que tal avaliação fará parte dos trabalhos futuros que podem vir a ser realizados com o término deste trabalho.

Devido à natureza exploratória do mesmo, o período de tempo para realização foi consideravelmente curto. Entretanto, como de caráter experimental, foi possível alcançar um resultado satisfatório. Uma das limitações encontradas para realizar este trabalho, além do tempo, foi a dificuldade para contatar os gerentes e coordenadores, pois muitos alegaram indisponibilidade devido a questões do trabalho. Por esse fato, a pesquisa sofreu uma defasagem quantitativa e qualitativa. Quantitativa, porque só foi possível entrevistar metade da amostra pretendida. E qualitativa, porque todos os que entrevistados trabalham em unidades da administração central (pró-

reitorias), não havendo nenhum diretor de centro acadêmico ou órgão suplementar da universidade, logo, os problemas que foram identificados derivam de um ‘foco específico’.

Para trabalhos futuros, visa-se expandir a pesquisa a toda a universidade, incluindo todos os campus, visitando todas as pró-reitorias, centros acadêmicos e órgãos suplementares, para obter um panorama geral da utilização do PAI, o que seria de grande valia para o desenvolvimento de novas práticas para a melhoria da Metodologia PAI da UFPE.

Referências

- ALENCAR, B. P. de; SOUZA, D. C. M. de. Manual de Gestão por Processos. Brasília, 2013.
- ALMEIDA, Egle Luís Vieira de. METODOLOGIA PAI: UMA APLICAÇÃO PARA ANDROID. 2017. 56 f. Trabalho de Graduação - Curso de Engenharia da Computação, Universidade Federal de Pernambuco, Recife, 2017.
- AMBLER, Scott W. Disciplined Agile Software Development: Definition. 2005. Disponível em <<http://www.agilemodeling.com/>>. Acessado em 18/10/17.
- ANICHE, Maurício. Test-Driven Development. 2014. Disponível em <<http://tdd.caelum.com.br>>. Acessado em 22/11/17.
- AWAD, M. A. (2005). A comparison between Agile and traditional software development methodologies. This report is submitted as partial fulfillment of the requirements for the Honours Programme of the School of Computer Science and software Engineering, The University of Western Australia.
- BECK, K. Extreme Programming explained: Embrace change. Reading, Mass., Addison-Wesley, Nov16, 2004.
- COHN, M. and FORD, D. (2003) Introducing an Agile Process to an Organization IEEE Computer June: 74- 78.
- DAVENPORT, Thomas H. Reengenharia de Processos. Rio de Janeiro: Campus, 1994.
- DREYFUSS, Cassio. As redes e a gestão das organizações. Rio de Janeiro: Guide, 1996.
- FONSECA, Augusto V. M. da. MIYAKE, Dario Ikuo. UMA ANÁLISE SOBRE O CICLO PDCA COMO UM MÉTODO PARA SOLUÇÃO DE PROBLEMAS DA QUALIDADE. 2006. 9 f. XXVI ENEGEP, Fortaleza, PE, Brasil, 9 a 11 de Outubro de 2006.
- GALBRAITH, Jay. Designing organizations. San Francisco: Jossey-Bass, 1995.

GARVIN, David. The processes of organization and management. Sloan Management Review, v. 39. N. 4. Summer 1998.

GIL, Antonio Carlos. Métodos e técnicas de pesquisa social. 5. ed. São Paulo: Atlas, 1999. P. 202. ISBN: 8522422702.

GONÇALVES, José Ernesto Lima. Características do trabalho no ambiente do escritório. Trabalho apresentado no Expomicro, São Paulo, julho de 1990.

HARRINGTON, H. James. Business Process Management. New York McGraw Hill, 1991.

HIGHSMITH, Jim. Agile Project Management: Creating Innovative Products. Pearson Education, Jul10, 2009.

MADI, T., DAHALIN, Z., & BAHAROM, F. (2011). Content analysis on agile values: A perception from software practitioners. In Proceedings of 2011 Malaysian Conference in Software Engineering, 423–428. IEEE. doi:10.1109/MySEC.2011.6140710

MARSHALL JUNIOR, I.; CIERCO, A. A.; ROCHA, A. V.; MOTA, E. B.; LEUSIN, S. Gestão da Qualidade. 8ª edição. Rio de Janeiro: FGV, 2006.

MONIRUZZAMAN, A. B. M.; HOSSAIN, Dr. Syed Akhter. Comparative Study on Agile software development methodologies.

NASCIMENTO, Allan Leandro Bezerra do. GESTÃO DE AÇÕES INSTITUCIONAIS BASEADA NA METODOLOGIA PAI: UMA EXPERIÊNCIA NO INSTITUTO FEDERAL DE PERNAMBUCO. 2016. 112 f. Dissertação (Mestrado) - Curso de Ciência da Computação, Universidade Federal de Pernambuco, Recife, 2016.

PAWAR, Rupali Pravinkumar. A Comparative study of Agile Software Development Methodology and traditional waterfall model. IOSR Journal of Computer Engineering (IOSR-JCE). Innovation in engineering science and technology (INCIEST-2015).

PALMER, S.R. and FELSING, J.M., A Practical Guide to Feature-Driven Development. Upper Saddle River, NJ, Prentice-Hall, 2002.

PFLIEGER, S. and JOANNE, A. 2006. Software Engineering: Theory and Practice. 3rd Edn. , Pearson Prentice Hill, New Jersey.

POPPENDIECK, Mary and POPPENDIECK, Tom. 2003. Lean Software Development – An Agile Toolkit. Pearson, 1st Edn.

PROPLAN. UFPE, Manual de Elaboração do Plano de Ação Institucional 2013. Recife, PE, dezembro de 2012.

PROPLAN. UFPE, Manual de Elaboração do Plano de Ação Institucional 2014. Recife, PE, dezembro de 2013.

PROPLAN. UFPE, Manual de Elaboração do Plano de Ação Institucional 2015. Recife, PE, dezembro de 2014.

PROPLAN. UFPE, Manual de Elaboração do Plano de Ação Institucional 2016. Recife, PE, dezembro de 2015.

PROPLAN. UFPE, Manual de Elaboração do Plano de Ação Institucional 2017. Recife, PE, dezembro de 2016.

PROPLAN. UFPE, Fluxo do processo de planejamento e monitoramento. Recife, PE, sem data.

QURESHI, M. R. J. (2012). Agile software development methodology for medium and large projects. IET Software, 6(4), 358–363.

RAMOS, João Victor Wanderley. Definição e Prototipação de uma Ferramenta de Apoio ao Gerenciamento do Plano de Ação Institucional da UFPE. 2012. 49 f. Trabalho de Graduação – Curso Ciência da Computação, Universidade Federal de Pernambuco, Recife, 2012.

RIBEIRO, Elisa Antônia. A perspectiva da entrevista na investigação quantitativa. Evidência: olhares e pesquisa em saberes educacionais, Araxá/MG, n. 04. p. 129-148, maio de 2008.

RISING, L. and JANOFF, N. S. The Scrum software development process for small teams, IEEE Software, Issue 17, pp. 26-32 , 2000

SCHWABER, K. and BEEDLE, M. 2001. Agile Software Development with Scrum. 1st Edn., Prentice Hall, New Jersey.

UFPE, Relatório do Plano de Ação Institucional 2015. Recife, PE, dezembro de 2014.

UFPE, Relatório do Plano de Ação Institucional 2016. Recife, PE, janeiro de 2017.

VIEIRA FILHO, Geraldo. Gestão da Qualidade Total: uma abordagem prática. Alínea, 2014. P.24-29.

Apêndice

Roteiro da entrevista:

1. Você trabalhou diretamente com a criação/desenvolvimento dos processos do PAI?
 - a. Caso sim: Poderia descrever como foi esse processo e como os mesmos são aplicados?
 - b. Caso não: Tem algum conhecimento sobre como os processos são aplicados?
2. Você acredita que todos os processos ocorrem de forma eficiente?
3. Existe algum processo que você considere deficiente/falho?
 - a. Caso sim: Poderia descrever qual(is) a(s) falha(s) identificada(s) nesse processo?
 - b. Caso não: Você reconhece que o processo precisa de melhoria?
4. A utilização do PAI tem auxiliado no gerenciamento dos processos da UFPE?

Em relação ao PDCA:

5. Com relação ao Planejamento, como o PAI tem beneficiado no planejamento da UFPE?
6. Com relação à Execução, como o PAI tem contribuído na execução das ações institucionais da UFPE?
7. Com relação à Verificação, a adoção do PAI tornou mais fácil gerenciar as ações institucionais da UFPE?
8. Com relação ao Agir, como o PAI tem ajudado no que diz respeito às ações corretivas?