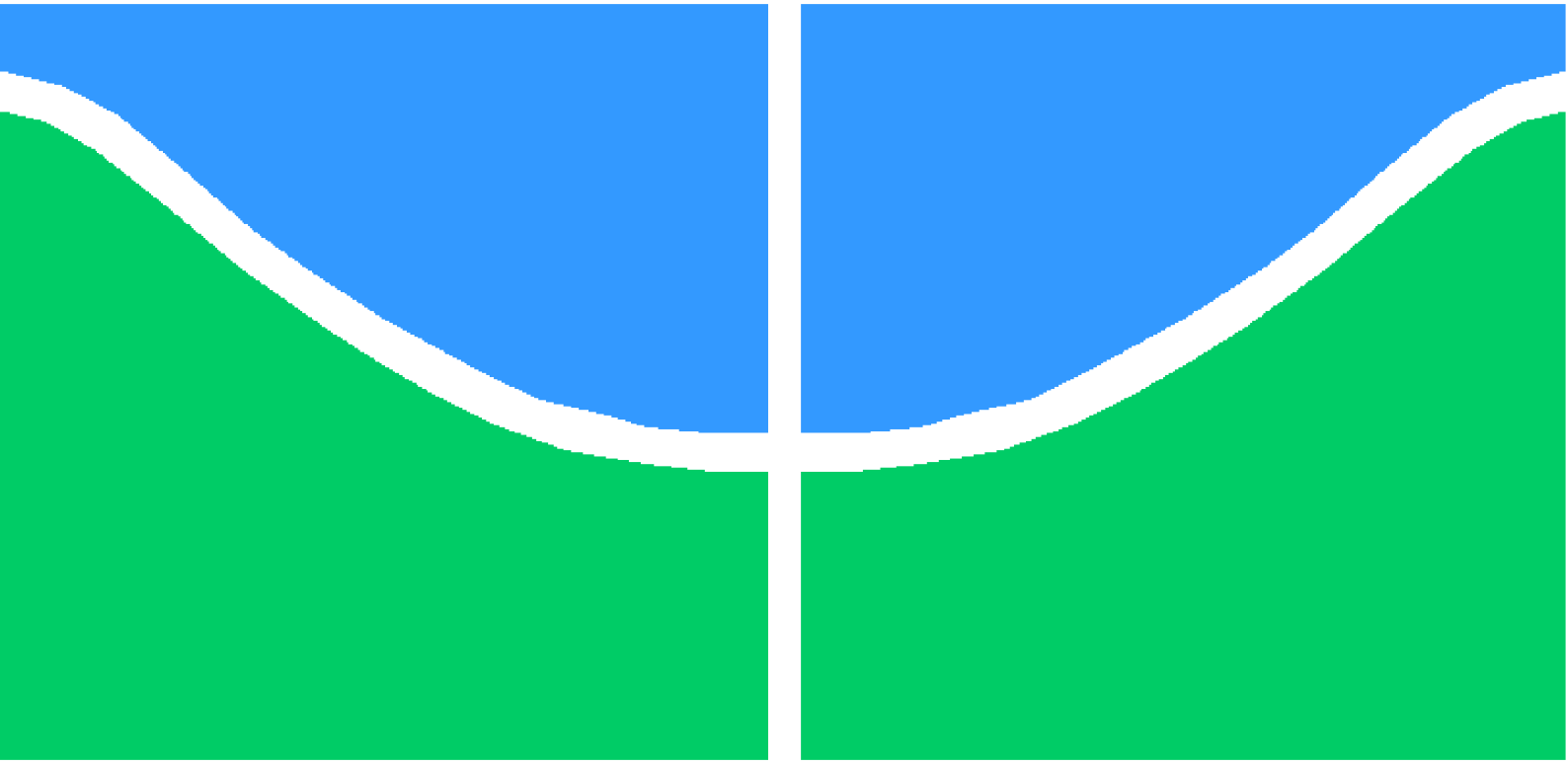

Universidade de Brasília - UnB
Faculdade UnB Gama - FGA
Engenharia de Software

Report sobre MPS

Autor: Karine Valencça
Murilo Duarte
Tiago Ribeiro de Assunção
Wilton Rodrigues
Professor: Dr. Edgard Costa Oliveira

Brasília, DF
2016

Karine Valencça
Murilo Duarte
Tiago Ribeiro de Assunção
Wilton Rodrigues

Report sobre MPS

Atividade submetida ao curso de graduação em (Engenharia de Software) da Universidade de Brasília, como requisito parcial para obtenção da aprovação em Melhoria de Processo de Software.

Universidade de Brasília - UnB

Faculdade UnB Gama - FGA

Orientador: Dr. Edgard Costa Oliveira

Brasília, DF

2016

Lista de abreviaturas e siglas

FS	Fábrica de *Software*
UnB	Universidade de Brasília
MPS	Melhoria de Processo de Software
MPS.BR	Melhoria de Processo do Software Brasileiro

Sumário

1 Documento de Visão

1.1 Introdução

Esse documento tem por objetivo fornecer uma visão geral da solução de software a ser empregada para dar suporte às atividades do produto *Bibliotech*. Aqui será descrito o problema a qual o software visa solucionar e dar apoio, o perfil do usuário que estará utilizando o sistema, e uma breve descrição de suas funcionalidades.

1.2 Posicionamento

Essa seção contém a descrição dos problemas, envolvidos e usuários, e subsequentemente a posição do produto.

1.3 Descrição dos Problemas

Nas figuras abaixo estão descritos os problemas identificados.

O problema de	Incapacidade do robô de identificar qual livro está sendo requisitado ou devolvido
afeta	O correto funcionamento do robô
cujo impacto é	O robô não identificar e pegar os livros solicitados
uma boa solução seria	Um sistema que identifique o livro e passe a informação para o robô de uma maneira que ele compreenda.

Figura 1 – Problema 01 (fonte: Autor)

O problema de	Administrar as informações dos livros da biblioteca
afeta	O conhecimento acerca dos livros mantidos
cujo impacto é	O descontrole sobre a administração dos livros
uma boa solução seria	Um sistema que permita manter e gerir as informações sobre os livros da biblioteca

Figura 2 – Problema 02 (fonte: Autor)

O problema de	Conhecer a disponibilidade física e informações de um livro
afeta	A capacidade de afirmar se um livro está disponível ou não
cujo impacto é	A execução da solicitação de retirada de um livro sem sucesso
uma boa solução seria	Um sistema que identifique o livro e informe a sua disponibilidade.

Figura 3 – Problema 03 (fonte: Autor)

O problema de	Identificar a relação entre livros emprestados e com quem esse livro está
afeta	A identificação do responsável pela devolução e integridade do livro
cujo impacto é	Perda da rastreabilidade de quem está com determinado livro emprestado
uma boa solução seria	Um sistema que mantenha os usuários e relacione eles com seus livros solicitados.

Figura 4 – Problema 04 (fonte: Autor)

1.4 Descrições dos Usuários

Essa seção apresenta uma descrição geral dos usuários do sistema, assim como suas respectivas responsabilidades e as suas necessidades ao utilizar o sistema.

1.4.1 Resumo dos Usuários

A a figura abaixo apresenta a descrição de cada usuário do sistema.

Nome	Descrição	Responsabilidades
Bibliotecário	É o usuário quem cadastra os livros no sistema e permite que o estudante solicite livros no sistema	- Inserir informações sobre os livros no sistema - Solicitar o empréstimo ou a devolução de um livro - Cadastrar estudantes no sistema
Estudante	É o usuário que está interessado em solicitar um livro na biblioteca	- Solicitar o empréstimo de um livro
Usuário	Generalização, dos atores, Bibliotecário e Estudante.	- Consultar livros

Figura 5 – Descrição dos usuários do sistema (fonte: Autor)

1.4.2 Principais Necessidades dos Usuários

A figura a seguir apresenta as necessidades dos usuários do sistema, a prioridade e quais são as soluções propostas para cada necessidade.

Necessidade	Prioridade	Soluções Propostas
Solicitar um livro (PB01)	Alta	Permitir ao BIBLIOTECÁRIO solicitar ao robô um livro a partir da identificação do mesmo junto ao sistema
Devolver um livro (PB01)	Alta	Permitir ao BIBLIOTECÁRIO devolver ao robô um livro a partir da identificação do mesmo junto ao sistema
Controlar as informações dos livros (PB02)	Alta	Permitir ao BIBLIOTECÁRIO realizar o cadastro, alterar, ou remover livros junto ao sistema
Saber a disponibilidade de um livro (PB03)	Media	Permitir ao BIBLIOTECÁRIO e ao ESTUDANTE realizar uma consulta da disponibilidade de um livro junto ao sistema
Controlar as informações dos estudantes (PB04)	Media	Permitir ao BIBLIOTECÁRIO realizar cadastro, alterar, ou remover o ESTUDANTE junto ao sistema
Acompanhar aquisição dos livros (PB04)	Media	Permitir ao BIBLIOTECÁRIO manter a rastreabilidade de quem possui determinado livro emprestado

Figura 6 – Necessidades dos usuários do sistema (fonte: Autor)

1.5 Visão Geral do Produto

Essa seção fornece o detalhamento de como o sistema se relaciona com os outros componentes do produto.

1.5.1 Perspectiva do Produto

O sistema descrito neste documento faz parte do produto *Bibliotech*. Dessa forma, ele é um subsistema e interage com outros subsistemas, a fim de atingir o objetivo final do produto.

A função do portal é implementar uma interface entre o usuário e o *software* embarcado que realiza o controle das funções do robô. Para isso, ele fornece uma interface *web* que permite ao bibliotecário solicitar que determinadas ações sejam executadas pelo robô. Em última instância, estas solicitações provocam a emissão de uma série de comandos para o robô.

A forma em que o sistema *web* se comunica com o *software* embarcado é através de uma rede *wireless* local. A solicitação da realização de uma tarefa feita pelo usuário gera uma requisição ao *software* embarcado que envia um código específico que indica ao *software* o tipo de tarefa que o robô deve realizar. Desta forma, o *software* embarcado é capaz de determinar quais funções do robô devem ser acionadas.

Após o término da tarefa do robô, o *software* embarcado envia uma resposta ao sistema *web* indicando o status da tarefa. Por fim, o sistema *web* fornece um *feedback* para o usuário indicando o status da tarefa conforme os padrões de usabilidade adotados no projeto.

2 Documento de Arquitetura de *Software*

Esta seção possui como finalidade apresentar uma visão abrangente da arquitetura dos *softwares* desenvolvidos para o projeto *Bibliotech*. Para isso, serão fornecidas uma série de visões arquiteturais para ilustrar os diversos aspectos do sistema, seus componentes e a forma em que interagem entre si.

2.1 Representação Arquitetural

O projeto *Bibliotech* possui uma série de *softwares* que realizam diferentes funções para garantir o funcionamento do sistema. Cada um destes *softwares* apresenta sua própria arquitetura interna, mas também estão inseridos em uma infra-estrutura externa projetada pela equipe para viabilizar a comunicação entre os subsistemas de *software* .

Serão descritos os aspectos da infra-estrutura a qual os subsistemas estão inseridos e posteriormente detalhar aspectos relacionados à arquitetura interna de cada um.

Em primeiro lugar, será especificado quais *softwares* compõe o sistema:

1. Portal da biblioteca: O portal da biblioteca é uma aplicação web que permite aos bibliotecários gerenciarem os seus livros de forma facilitada. Ou seja, este subsistema implementa a digitalização do empréstimo, adição ao acervo e devolução de livros. O portal abstrai para o usuário todo o processo de gerência de armazenamento implementado logicamente por ele e fisicamente pelo robô. Ele define internamente a administração do uso das prateleiras, por exemplo, quais são as posições livres da prateleira e onde um determinado livro se encontra. Este portal será desenvolvido com o *framework* web Rails.

2. Sistema Embarcado: Este subsistema é embarcado em uma *Raspberry Pi* e implementa o controle do robô, ou seja, emite uma série comandos que especificam ao robô as operações que devem ser executadas. As principais operações suportadas pelo robô são a movimentação horizontal e vertical (para acessar qualquer parte da estante) e a extração e reposição de um livro a uma posição da estante. Além disso, este subsistema se comunica com o portal da biblioteca para aceitar requisições que especificam a execução de uma função específica (por exemplo, armazenar um livro). Estas requisições disparam a emissão de uma série de comandos internos ao robô para que o mesmo cumpra a função solicitada. O sistema embarcados será desenvolvido em linguagem C.

A figura abaixo fornece uma descrição visual dos *softwares* que compõem o sistema e suas interações.

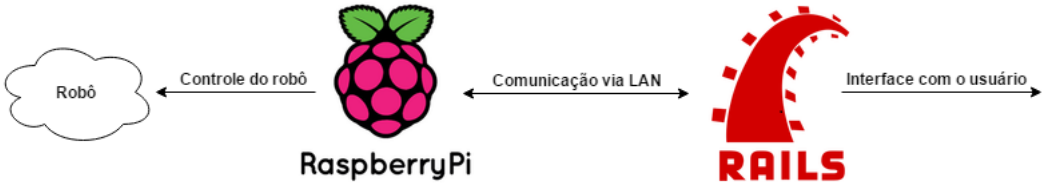

Figura 7 – Arquitetura do sistema (fonte: Autor)

O fluxo de execução do sistema pode ser resumido a seguir:

1. Portal da biblioteca: O portal da biblioteca é uma aplicação web que permite aos bibliotecários gerenciarem os seus livros de forma facilitada. Ou seja, este subsistema implementa a digitalização do empréstimo, adição ao acervo e devolução de livros. O portal abstrai para o usuário todo o processo de gerência de armazenamento implementado logicamente por ele e fisicamente pelo robô. Ele define internamente a administração do uso das prateleiras, por exemplo, quais são as posições livres da prateleira e onde um determinado livro se encontra. Este portal será desenvolvido com o *framework* web Rails.O bibliotecário interage com a interface web para realizar algum tipo de atividade administrativa (adicionar um livro ao acervo, devolução de livros, empréstimo, etc...). Cada uma dessas atividades administrativas exige um gerenciamento interno do sistema web para identificar o estado das estantes (quais posições estão sendo utilizadas, por exemplo) e dos livros (quais livros estão disponíveis e onde estão armazenados). Para isso, a aplicação web contará com uma base de dados que será responsável por armazenar estas informações. Por fim, o portal da biblioteca enviará requisições ao *software* sendo executado na *Raspberry Pi* informando algum tipo de ação a ser executada.

2. O servidor sendo executado na *Raspberry Pi* aceitará requisições do portal da biblioteca e mapeará uma requisição a uma série de comandos que especificam ao robô as ações que devem ser tomadas para cumprir a função solicitada.

3. As funções do robô são acionadas a partir do *software* embarcado na *Raspberry Pi* e após seu término, o servidor envia uma resposta ao portal da biblioteca notificando a conclusão da solicitação enviada.

4. O portal informa ao usuário que a função solicitada foi executada.

2.2 Metas e Restrições da Arquitetura

Existem algumas metas e restrições arquiteturais consideradas relevantes pela equipe de desenvolvimento, são elas:

1. A aplicação web será executada em uma rede local, pois não faz sentido a atribuição de um IP público se ela só será acessada pelos funcionários da biblioteca e se comunicará com o servidor da *Raspberry Pi* que também pertence à rede local.

2. O portal da biblioteca deve apresentar um sistema de identificação e autenticação que garanta que apenas funcionários autorizados possam acessá-lo com sucesso.

3. O servidor da *Raspberry Pi* deve aceitar requisições exclusivamente do portal da biblioteca.

4. A implantação do sistema em uma organização deve ser automatizada, para evitar a configuração manual de IPs para viabilizar a comunicação entre os subsistemas de *software*.

2.3 Visão Lógica

Esta seção se concentra em descrever detalhes arquiteturais internos de cada subsistema:

1. Sistema Web: O sistema web será implementado utilizando o estilo arquitetural MVC (*Model*, *View*, *Controller*). Este padrão arquitetural determina a subdivisão do *software* em camadas com responsabilidades específicas:

   Model: responsável por representar as entidades envolvidas no projeto e implementar a interface entre o sistema e a base de dados.

   View: responsável por conter o código relacionada à camada visual da aplicação.

   Controller: responsável por coordenar o fluxo de execução do sistema e atuar como a interface entre as camadas descritas anteriormente.

   A aplicação web também será responsável por administrar o uso das estantes. Para isso, o projeto implementa um recurso de discretização das estantes. A discretização é possibilitada pela construção de cases de tamanho único que irão envolver os livros, dessa forma, o espaço ocupado por cada livro é determinístico e o *software* é capaz de interpretar a estante como uma matriz binária, onde cada posição da estante pode ser representada por um par ordenado. Este par ordenado será armazenado para identificar a posição exata de um determinado livro.

2. Sistema Embarcado: O sistema embarcado será dividido em dois módulos principais:

   Servidor: responsável por aceitar requisições do sistema web e processá-las para identificar quais comandos devem ser enviados para o robô, além de enviar mensagens de resposta informando ao portal o estado da execução da tarefa solicitada.

   Controlador: responsável por enviar comandos para o robô para acionar suas funções. Cada função do robô está associada a um dispositivo externo, por exemplo, um motor de passo. Estes dispositivos são representados internamente por um arquivo de dispositivo. A comunicação entre o *software* e os dispositivos periféricos será realizada através destes arquivos e da interface de programação unificada fornecida por sistemas Unix-like para manipulação de arquivos.

3 Gerenciamento da Comunicação e Versões de Artefatos

3.1 Ferramentas de Comunicação

Objetivando facilitar a troca de informações entre os membros da equipe do projeto, foram escolhidas algumas ferramentas para auxílio de comunicação e controle de artefatos e anexos do projeto.

Tendo em vista a importância e relevância de uma boa comunicação e gestão de artefatos em projetos interdisciplinares, foram selecionados mecanismos gratuitos para garantir que todos os membros do projeto tivessem acesso livre de forma espontânea.

A seguir estão apresentadas as ferramentas de comunicação e gestão adotadas e seus propósitos:

Ferramenta	Propósito
hline Facebook	Espaço escolhido para se disponibilizar links, arquivos, discussões sobre a org
hline WhatsApp	Comunicação básica e planejamento de re
hline Hangouts	Reuniões não presenciais
hline Google Drive	Compartilhamento e Edição de Arqui
hline	

3.2 Reuniões

Além das ferramentas apresentadas, foram definidas reuniões fixas e presenciais para facilitar a comunicação e execução de atividades do projeto, nos seguintes dias e horários:

Quarta-Feira: 16h às 18h Sexta-Feira: 14h Às 18h

Além disso, algumas reuniões extras poderão ser realizadas caso a equipe considere necessário. Contudo, essas reuniões devem ser agendadas de forma democrática com, no mínimo, 2 dias de antecedência, pretendendo a maior presença dos membros do equipe e/ou membros que se façam necessários.

4 Introdução

Este presente trabalho tem como objetivo discutir sobre alguns pontos levantados durante a aula de Melhoria de Processo de Software. Com auxílio do orientador da disciplina, descorreremos sobre como a melhoria de processo pode influenciar na execução das atividades, bem como elas acontecem.

4.1 Questões de Pesquisa

A fim de nos auxiliar a compreender melhor o processo de melhoria de software, nosso orientador propos cinco perguntas. E para bem entender o contexto, iremos respondê-las ao longo deste trabalho:

1. A melhoria dos processos de desenvolvimento de software pode ajudar a diminuir o impacto de algumas das dificudades essenciais descritas por Brooks? E as acidentais?
2. Dê um exemplo de processo descrevendo para ele: entrada, saída, atividades, papel, política, ferramenta e método.
3. Descreva um cenário de trabalho onde é utilizado um processo imaturo e outro cenário onde é utilizado um processo maduro.
4. Descreva uma atividade de cada fase do processo de implantação de MPS.
5. Descreva uma atividade de cada fase do processo de avaliação de processos.

5 Desenvolvimento

Nesta sessão iremos responder todas as questões propostas anteriormente, sendo que cada subsessão será responsável pela sua respectiva questão.

5.1 Primeira Questão

Certamente, uma vez que ao se melhorar um processo defeituoso, ou em alguns casos estabelecer um processo formal que até então inexistente, vários problemas de execução e gargalos no fluxo do processo são resolvidos. No caso das dificuldades essenciais, que são aquelas pertinentes à natureza do software, uma melhoria no processo na fase de elicitação de requisitos pode prevenir que ocorram lacunas nessa tarefa que possam comprometer o produto final de software, fazendo com que o mesmo não esteja alinhado com objetivo de negócio do cliente. Já com relação às dificuldades, uma vez que o processo esteja bem amadurecido o impacto nesse tipo de dificuldade é bastante notável, pois com papéis e tarefas bem definidas os riscos (dificuldades) como problemas com tecnologias, ferramentas, linguagens e assim por diante, são facilmente percebidos e consequentemente mitigados.

5.2 Segunda Questão

Um exemplo de processo muito comum para o ambiente de desenvolvimento de software é a implementação de uma história de usuário. Esta, como entrada, contém o requisito solicitado pelo usuário e como saída, o incremento de software produzido. A atividade é desempenhada por aquele que tem o papel de desenvolvedor, executando assim, a implementação do requisito.

A política utilizada é que o desenvolvedor só poderá declarar a história como concluída quando todos os critérios de aceitação estiverem cumpridos, como: cobertura de testes, enquadramento em folha de estilo, entre outros pontos. Como método, o desenvolvedor utilizou metodologia ágil.

5.3 Terceira Questão

Um exemplo de processo imaturo é o cenário onde um desenvolvedor recebe o requisito do cliente e o implementa, fazendo-o de maneira cíclica e sem priorização. Neste caso, não existe nenhum controle sobre as saídas ou entradas do processo, apenas é feito sem organização.

Um processo maduro, como exemplo, é o produto que as disciplinas de MDS/GPP desenvolvem durante o semestre, onde é feito o planejamento das entregas, medições dos resultados, dos incrementos de software.

5.4 Quarta Questão

Nesta sessão, serão abordadas as fases para que seja possível fazer a implantação de um projeto de Melhoria de Projeto de Software. Temos as seguintes fases a serem pontuadas:

1. Iniciação
2. Diagnóstico
3. Planejamento
4. Ação
5. Aprendizado

Todos esses pontos serão discutidos nas subsessões a seguir:

5.4.1 Iniciação

Nessa fase, é necessário que haja a escolha de qual contexto será aplicada a melhoria. Bem como quais são os interessados nela. Pois se não houver nenhum patrocinador, não é possível executar o processo a fim de melhorá-lho. Dessa maneira, há a necessidade de uma pessoa responsável por bancar a produção

5.4.2 Diagnóstico

Assim que haja fundos e recursos para investir na melhoria do projeto, a melhori tem que ser implementada. E para isso, alguns passos são necessários, dentre eles, a avaliação do produto do projeto por meio de um diagnóstico. E com essa fase de análise, já podem ser levantados pontos para executar a sugestão de melhoria, ou recomendações para tal.

5.4.3 Planejamento

Intuitivamente, após todos os pontos serem levantados, há a necessidade de se planejar sobre como será implementada as mudanças no projeto, bem como definir quais são os principais pontos da mudança e desenvolver um plano de implantação.

5.4.4 Ação

Depois de analisado, sugerido e planejado, chega a desejada hora de aplicar as mudanças no projeto, fazendo um teste piloto, após, refinando a solução e, por fim, implementando-a de fato.

5.4.5 Aprendizado

Após todo processo de implantação, é feita a colheita de resultados, analisando e avaliando o quão eficaz aquelas sujestões de melhoria foram para o projeto. E se obtiveram êxito ou não.

Analisando os resultados finais, a equipe consegue então verificar quais melhorias podem ser propostas para o projeto. Dessa maneira, faz-se um ciclo de melhoria contínua onde os resultados do aprendizado servem de entrada para o diagnóstico do processo, possibilitando assim que o desenvolvimento do software esteja em constante melhoria.

5.5 Quinta Questão

Uma avaliação de processo de software pode ser dividida em 3 grandes fases, compostas por tarefas menores. Na primeira fase, a fase de Planejar e preparar a avaliação, uma das tarefas mais importantes é a de analisar requisitos, pois a partir dela é possível obter um entendimento inicial do projeto para ser possível identificar os pontos aos quais o projeto se propões a resolver. A segunda fase, Condução da avaliação, mais expecificamente ao fim desta fase, na tarefa de gerar resultados da avaliação é onde após todos os levantamentos terem sido feitos que obteremos a avaliação propriamente dita. E por fim, a fase de Relatar os resultados é onde a avaliação obtida a partir da fase anterior será entregue e posteriormente arquivada.