

Plano de Projeto do Sistema de Gestão
Eletrônica de Documentos da Defensoria
Pública do Distrito Federal - SGED/DPDF

Pedro Salum
09/0139232
pedro@loopec.com.br

Daniel Sandoval
09/0109899
daniel@loopec.com.br

3 de março de 2013

Sumário

1	Introdução	2
2	Processo de Desenvolvimento	3
3	Arquitetura e padrões de projeto	5
4	Plano de Testes	7
5	Plano de Implantação e Manutenção	8
6	Análise de Riscos	9

Capítulo 1

Introdução

Este documento tem por objetivo explicitar o planejamento para a realização do projeto de Gestão Eletrônica de Documentos da Defensoria Pública do Distrito Federal — SGED/DPDF — no tocante ao desenvolvimento, arquitetura, testes, implantação, manutenção e riscos. A seguir há uma breve descrição de cada capítulo.

Processo de Desenvolvimento Discorre sobre decisões tomadas em relação a metodologias de desenvolvimento de *software* e gerência de projeto. Discute a escolha de metodologias ágeis sobre métodos planejados.

Arquitetura e padrões de projeto Discorre sobre escolhas de arquitetura e padrões de projeto que serão utilizados na codificação do sistema.

Plano de Testes Discorre sobre as metodologias e ferramentas de testes escolhidas para o projeto, considerando desde testes de código até de aceitação, como e quando serão realizados.

Plano de Implantação e Manutenção Discorre sobre as metodologias e ferramentas escolhidas para o processo de implantação e manutenção contínua do sistema entregue.

Análise de Riscos Discorre sobre riscos que devem ser considerados ao longo do projeto e possibilidades de mitigação.

Capítulo 2

Processo de Desenvolvimento

A metodologia de desenvolvimento escolhida foi baseada em métodos ágeis. Levando-se em consideração que o sistema a ser desenvolvido está inserido em um ambiente corporativo, temos que estar preparados para mudanças. Mudanças de prioridades; regras de negócio; gerência; cronograma; entre outras, devem ser suportadas com o mínimo de prejuízo possível. Além disso, há experiências bem sucedidas com desenvolvimento ágil em ambientes corporativos governamentais [3].

Sabendo que métodos ágeis não são próprios para implantação uniforme em qualquer situação [1], as ferramentas e metodologias escolhidas para este projeto são uma sugestão e podem sofrer modificações ou adaptações ao longo do desenvolvimento.

Scrum Metodologia de gerência de projetos, tem sido muito utilizada para desenvolvimento de software de maneira ágil [4]. As iterações utilizadas (*sprints*) terão duração de duas semanas, sendo que uma versão funcional do software tem que ser entregue ao final de cada *sprint*.

As funcionalidades serão divididas em *user stories* e catalogadas no *product backlog*. Ao início de cada *sprint*, algumas serão selecionadas e movidas para o *sprint backlog*, tendo de ser implementadas até o final da *sprint*.

TDD — Test-driven Development Metodologia que aumenta qualidade de código através da implementação de testes antes da implementação da funcionalidade que será testada. O uso de TDD no desenvolvimento reduz significativamente o número de defeitos com pouco *overhead* [2].

Versionamento O versionamento será intensamente utilizado para enfatizar todas as políticas de desenvolvimento planejadas. A ferramenta escolhida

foi o `git` [5], por sua notória excelência em versionamento. Ao início de cada *sprint*, deve ser criado um *branch* para cada *user story* e, de cada *branch*, um para cada desenvolvedor. Ao final do dia, todos os *branches* devem ser *merged* no da *sprint*.

Capítulo 3

Arquitetura e padrões de projeto

O sistema a ser desenvolvido deve ser completamente Web, para que seja acessível de qualquer estação de trabalho da DPDF ou qualquer computador ou dispositivo móvel ligado à Internet sem configuração necessária. Para sistemas Web, a arquitetura natural é a MVC (*Model View Controller*) e será utilizada extensivamente. Adicionalmente, há outro desafio, devido à natureza distribuída da Instituição, cada Núcleo de Atendimento deve funcionar de maneira independente caso não haja conexão com a Sede. Além disso, as informações devem manter-se sincronizadas entre os Núcleos e a Sede. Para contemplar essa característica, será adotada uma arquitetura *peer-to-peer*.

MVC Consolidada arquitetura para aplicações que variam de aplicativos móveis a sistemas web, ela provê separação entre o código responsável pela lógica de negócio e o código responsável pela apresentação e interface com o usuário.

Peer-to-peer Para permitir alta disponibilidade, a arquitetura *peer-to-peer* será utilizada, distribuindo processamento e dados pelos Núcleos de Atendimento e Sede.

Padrões de Projeto

Singleton Será utilizado para prestadores de serviço, como serviço de autenticação, por exemplo. Através da utilização desse padrão de projeto, podemos centralizar operações lógicas aumentando a confiabilidade no cumprimento das regras de negócio.

Observer Para operações encadeadas ou acessórias será utilizado o esse padrão, possivelmente inclusive através de um *framework* de **AOP** — **Aspect-**

oriented Programming, permitindo código mais limpo e sucinto, focado na lógica de negócio.

State Padrão de projeto bastante útil para autenticação, diferenciando um usuário entre os estados de “autenticado”, “anônimo” ou inclusive delineando os papéis assumidos por aquele usuário.

Capítulo 4

Plano de Testes

Baseado no desenvolvimento ágil e na ideia de manter o código coeso e funcional, dividimos o teste em 3 partes:

Unitário O teste unitário é efetuado continuamente pelos desenvolvedores, ou seja, após a compilação do código. Todos os códigos são devidamente testados utilizando TDD e só são dados *push* após passarem nos testes que foram previamente testados. Os desenvolvedores trabalham em seus *branches* separadamente os quais são juntados ao “master” somente pela equipe de integração.

Integração O teste de integração é efetuado por uma equipe de desenvolvedores, onde todos os *branches* são unidos no final de cada dia de desenvolvimento, e verifica-se se existe algum erro de integração. Caso algum erro seja encontrado, o mesmo deve ser arrumado assim que o possível pela equipe de integração. Caso não seja possível arrumar, o pedaço de software é devolvido ao seu desenvolvedor, o qual será responsável por arrumar este erro.

Aceitação Efetua-se o teste de aceitação no final de cada *sprint* para verificar a aderência ao processo do órgão em questão. Este teste é efetuado por uma equipe mista, ou seja, composta dos Gerentes do Projeto, bem como a CTIC (Comitê de Tecnologia da Informação da DPDF). Caso algum erro conceitual seja encontrado, os requisitos do mesmo deverão ser verificados. Caso seja um erro de programação, estes deverão ser reportados à equipe de desenvolvimento.

Capítulo 5

Plano de Implantação e Manutenção

Implantação A implantação do sistema aqui descrito será por meio de um *deploy* em um servidor central da Defensoria Pública do Distrito Federal e uma cópia do mesmo em cada um dos 26 Núcleos de Assistência Jurídica espalhados pelo Distrito Federal. A cópia dentro do servidor central será configurada como “MASTER”, ou seja, recebe informações das outras 26 cópias espalhadas, configuradas como “SLAVE”, e as armazena para o próximo *backup*.

Manutenção Manutenções serão efetuadas em ambiente de teste, as quais após estarem estáveis e prontas para *deploy*, serão mandadas para o servidor central, substituindo a cópia “MASTER”, onde os “SLAVES” irão solicitar a versão mais nova do sistema o mais cedo possível, ou seja, uma cópia do “MASTER” será baixada para o “SLAVE” a qual substituirá a versão anterior. Importante ressaltar que só serão transmitidos as partes atualizadas do sistema, reduzindo a carga da rede.

Capítulo 6

Análise de Riscos

A pesar de ser um software consideravelmente pequeno, quando falamos se sistemas corporativos, ainda existem riscos que devem ser levados em consideração, mesmo que estes sejam completamente improváveis de acontecerem no curto-médio prazo. A seguir estão listados, em nível de probabilidade, e detalhados os riscos aqui levados em consideração.

Alta Eventos de alta probabilidade são aqueles referentes à equipe de desenvolvimento e suas relações com as partes, tanto UnB quanto DPDF. Dentre estes riscos estão:

- Saída de membro da gerência do projeto
- Saída de membro altamente qualificado, tanto na área de teste quanto de desenvolvimento
- Atraso no recebimento do capital para pagamento da equipe
- Falta de cooperação entre a Defensoria Pública do Distrito Federal e a equipe interna.

Para evitarmos problemas internos da equipe, iremos bem remunerá-los, manter reservas financeiras provenientes do não pagamento dos coordenadores no início do projeto, remunerando-os no final, gerando um caixa de segurança. Além da parte financeira, iremos prover toda a estrutura necessária para que o trabalho possa ser efetuado além de ambientes para descanso e descontração. Para evitar a falta de cooperação da equipe interna da DPDF, iremos negociar com a diretoria do órgão para que um membro seja alocado somente para se comunicar com a nossa equipe, retirando os outros afazeres do mesmo.

Média Eventos de média probabilidade são aqueles internos à instituição e que são possíveis de ocorrer e que podem impedir ou dificultar o desenvolvimento do sistema como um todo. Estes são:

- Cancelamento do Contrato entre DPDF e a Universidade de Brasília
- Troca do quadro de diretores
- Problemas de infraestrutura na área de TI, sejam com servidores ou com a parte de telecomunicação

Visando a redução destes riscos de média probabilidade, devemos efetuar a prestação de serviço de desenvolvimento de software o mais rápido possível, reduzindo os riscos de existir uma possível troca de gestão e cancelamento de contrato entre as partes. Sobre problemas de infraestrutura de TI, o sistema deverá funcionar tanto de forma local (Por meio de uma LAN) quanto em rede (WAN). Sendo assim, devemos partir para uma abordagem distribuída, com possível topologia de Malha ou Estrela estendida.

Baixa Eventos de baixa probabilidade são aqueles que modificam estruturas que são externas à Defensoria Pública do Distrito Federal, e não cabe à instituição decidi-las. Dentre elas estão:

- Modificação na estrutura judiciária do Distrito Federal
- Modificação de como ocorre a comunicação entre as partes (DPDF - Tribunal)

Com o objetivo de reduzir possíveis danos gerados por estes riscos, o sistema deve ser flexível o suficiente para se adaptar à tais mudanças, seja por meio de poucas modificações ou, idealmente, nenhuma modificação.

Muito Baixa Eventos de probabilidade baixa podem ser considerados catastróficos, ou seja, não só inviabilizam o desenvolvimento do software como impossibilitam o seu uso e existência da própria instituição. Entre eles estão:

- Terremoto com epicentro próximo à Brasília devasta toda a estrutura jurídica de Brasília e todo o recurso é devidamente realocado
- Meteoro atinge o prédio da DPDF
- Godzilla surge do fundo do Lago Paranoá e derruba o prédio da DPDF
- ETs invadem a terra e sugam o prédio da DPDF

Devido a baixa probabilidade de ocorrência dos riscos de importância Muito Baixa, e são impossíveis de serem mitigados, estes serão descartados para fins deste projeto.

Referências Bibliográficas

- [1] M. Cohn. *Succeeding with Agile: Software Development Using Scrum*. Pearson Education, 2009.
- [2] E.M. Maximilien and L. Williams. Assessing test-driven development at ibm. In *Software Engineering, 2003. Proceedings. 25th International Conference on*, pages 564 – 569, may 2003.
- [3] Claudia de O Melo and Gisele RM Ferreira. Adoção de métodos ágeis em uma instituição pública de grande porte-um estudo de caso. *Mensagem do Coordenador de Programa do WBMA*, page 112, 2010.
- [4] L. Rising and N.S. Janoff. The scrum software development process for small teams. *Software, IEEE*, 17(4):26 –32, jul/aug 2000.
- [5] Diomidis Spinellis. Git. *Software, IEEE*, 29(3):100 –101, may-june 2012.