

## **GDD Desde Dentro: mitos e realidades**

RELATÓRIO DE ESTÁGIO DE MESTRADO

**Valeria Vition**

MESTRADO EM ENGENHARIA INFORMÁTICA



UNIVERSIDADE da MADEIRA

*A Nossa Universidade*

[www.uma.pt](http://www.uma.pt)

setembro | 2016



**GDD Desde Dentro:**  
mitos e realidades

RELATÓRIO DE ESTÁGIO DE MESTRADO

**Valeria Vition**

MESTRADO EM ENGENHARIA INFORMÁTICA

ORIENTADOR  
Eduardo Leopoldo Fermé





---

# GDD DESDE DENTRO: MITOS E REALIDADES

---

**Valeria Vition**

(Licenciada)

*Relatório de Estágio Submetido à Universidade da Madeira para a  
Obtenção do Grau de Mestre em Engenharia Informática*

Funchal, Setembro de 2016



## **Orientador:**

Eduardo Leopoldo Fermé, PhD

*Professor Associado do C.C. de Ciências Exactas e da Engenharia da Universidade da Madeira*



# Abstract

While GDD (Global Development and Delivery) is becoming a way of life, this thesis represents a wide range of conclusions. However, GDD is technologically and organizationally complex and presents a variety of challenges to be managed by the software development team. In particular, temporal, geographical and socio-cultural distances impose problems not experienced in traditional systems development. In this paper, I present findings from an internship on GlobalSoftware company as a QA and two month lately as a PM in which I explore the particular challenges associated with managing GDD.

This Master thesis is exploring a lot of aspects of the global development and delivery regarding the software development. It is the coordination of development activities in places to produce software applications and management through distributed repositories of the assets that contribute to these applications (Lanubile 2009).

The purpose of this report is to go through the GDD, to understand the methodology, the research techniques used and the impact that this has within the organization in which I applied for. All the observations come from the experiences of working in a real background. This guides the development and delivery of software and also serves to study the patterns of myths and their interpretation in a GDD environment.

It has been revealed that the geographically distributed enterprise is not easy to be managed and needs more resources allocated. The purpose is the same, what changes is the way how we reach those goals. To understand the "intense" why, a physical presence was needed to observe the organization's characteristics and analyze the scenario of globalization.

This thesis is a step to a better, more productive and with a higher quality GDD. It helps companies to take into account and address potential challenges at the beginning by the overall structure. The presented work also helps companies to find the solutions validated to meet the challenges in their practice. At the same time, I have accumulated a considerable knowledge and experience, which is beginning to have a say in the future.

# Keyword

Global Development and Delivery

Project Management

Distributed Teams

Global Projects

Quality Assurance

Software Engineering



# Resumo

Enquanto GDD (Global Development and Delivery) torna-se um modo de vida, esse trabalho apresenta uma ampla gama de conclusões. No entanto, o GDD é tecnologicamente e organizacionalmente complexo e apresenta uma variedade de desafios a serem geridos pela equipa de desenvolvimento de *software*. Em particular, as distâncias temporais, geográficas e socioculturais impõem problemas não experimentados no desenvolvimento de sistemas tradicionais. Neste relatório, apresento as conclusões deduzidas durante o estágio na empresa GlobalSoftware, onde comecei por assumir funções de QA mas passado 2 meses comecei a exercer a função de PM explorando os desafios particulares associados á gestão de GDD.

Esta tese explora vários aspetos do desenvolvimento global e entrega no que se refere ao desenvolvimento de *software*. É a coordenação de atividades de desenvolvimento em locais para produzir aplicativos de *software*, bem como a gestão através de repositórios distribuídos dos ativos que contribuem para essas aplicações (Lanubile 2009).

O objetivo deste relatório de estágio passa por refletir sobre o processo global e de perceber através da metodologia e técnicas de investigação utilizadas, o impacto que este meio representa dentro da organização em que me inseri. As observações são provenientes da experiência de trabalhar num espaço real.

Este trabalho direciona o desenvolvimento e entrega de *software* e também serve para estudar os padrões dos mitos e a interpretação dos mesmos num ambiente GDD.

Tem-se revelado que a empresa geograficamente distribuída não é fácil de ser gerida e necessita de mais recursos atribuídos. Ou seja, a finalidade é a mesma, o que muda é a forma como se atingem esses objetivos. Para compreender o porquê “intenso”, foi

necessária uma presença física para perceber as características da organização e analisar o cenário de globalização.

Esta tese é um passo para uma GDD melhor, mais produtivo e com mais qualidade uma vez que ajuda as empresas a ter em conta e abordar desafios potenciais no início por meio da estrutura global. O trabalho apresentado também ajuda as empresas a encontrar soluções validadas para enfrentar os desafios em sua prática. Ao mesmo tempo, tenho acumulado um conhecimento considerável e uma experiência que está começando de uma forma abrangente.

# Palavras-chave

Desenvolvimento global e entrega

Gestão de Projetos

Equipas distribuídas

Projetos Globais

Garantia de Qualidade

Engenharia de Software



# Agradecimentos

O espaço limitado desta secção de agradecimentos, não me permite agradecer a todos que, ao longo do meu percurso me ajudaram a cumprir os meus objectivos e a realizar mais esta etapa da minha formação académica. Desta forma, deixo apenas algumas palavras, poucas, mas um sentido e profundo sentimento de reconhecido agradecimento.

Principalmente gostaria de expressar a minha sincera e profunda gratidão ao meu orientador Prof. Doutor Eduardo Leopoldo Fermé, do Centro de Competência de Ciências Exatas e da Engenharia da Universidade da Madeira, pela sua excelente orientação e apoio que muito elevaram os meus conhecimentos científicos e, sem dúvida, estimularam o meu desejo de sempre saber mais e a vontade constante de querer fazer melhor e que tornou possível a concretização da tese e a conclusão do mestrado.

Dirijo um agradecimento especial à minha família, pela incentivo, compreensão e encorajamento, durante todo este período, sem eles não chegava a este platamar.

Às equipas GlobalSoftware do Funchal e da central Londres pela oportunidade, por me terem recebido tão bem, pelo profissionalismo e por fazerem parte da minha viagem no mundo do trabalho.

À minha professora Maria do Céu Silva pela disponibilidade e apoio durante todos estes anos como professora e depois como amiga. Fico agradecida pelo acompanhamento e voto de confiança transmitido.

Agradecimentos a todos que fizeram parte da minha trajetória universitária e espero que esta etapa, que agora termino, possa, de alguma forma, retribuir e compensar todo o apoio e dedicação que, constantemente, me oferecem.

A eles, dedico todo este trabalho.



# Índice

<b>1</b>	<b>Introdução .....</b>	<b>1</b>
1.1	Motivação .....	3
<b>2</b>	<b>GDD .....</b>	<b>5</b>
2.1	Coordenação e Colaboração entre as equipas globalmente distribuidas.....	7
2.2	Benefícios GDD .....	12
2.2.1	Redução de custos.....	12
2.2.2	O acesso a grandes forças de trabalho multiqualificado.....	13
2.2.3	A redução Time to Market.....	13
2.2.4	A aproximação do mercado e dos clientes.....	13
2.2.5	Inovação e partilha de práticas melhores.....	14
2.2.6	Alocação de recursos.....	14
2.2.7	Melhorar a modularização das tarefas.....	14
2.2.8	Custos de coordenação baixos.....	15
2.2.9	Autonomia das equipas.....	15
2.2.10	Comunicação formal.....	15
2.2.11	Documentação melhorada.....	16
2.2.12	Processos claramente definidos.....	16
<b>3</b>	<b>Projeto Desenvolvido .....</b>	<b>17</b>
3.1	Contexto do estágio.....	20
3.2	Desafios em gestão de projetos distribuídos .....	22
3.2.1	Assumir funções QA.....	24
3.2.2	Assumir funções PM.....	26
3.3	A Organização .....	28
3.3.1	Fatores Relevantes na Divisão do Esforço do Desenvolvimento.....	28
3.3.2	A Gestão de Vários Componentes .....	29

3.3.3	A Integração .....	30
3.3.4	O projeto SoftwareX.....	30
<b>4</b>	<b>GDD: Mitos e Realidades .....</b>	<b>41</b>
4.1	Mitos principais .....	42
4.2	Outros Mitos.....	78
4.3	Mitos sobre os processos de modelagem.....	82
4.4	Mitos e Realidades vividas dentro do estágio.....	89
<b>5</b>	<b>Conclusão.....</b>	<b>91</b>
5.1	Visão Crítica do Estágio .....	91
5.2	Conclusão e Contribuição.....	95
5.3	Trabalhos Futuros.....	96
<b>6</b>	<b>Referências.....</b>	<b>97</b>

## Lista de figuras

Figura 1. Mapa do desenvolvimento do projeto GDD.....	6
Figura 2. Comunicar no ambiente GDD.....	8
Figura 3. Coordenação e Colaboração (Kotlarsky 2002).....	10
Figura 4. Plataformas Smart TV.....	32
Figura 5. Architecture - Browser Based TV.....	34
Figura 6. VOD Content with OpenStream.....	35
Figura 7. Content Workflow.....	36
Figura 8. Mapa de Gantt SoftwareX.....	37
Figura 9. Diagrama User Flow.....	38
Figura 10. Diagrama casos de uso ( <i>Login</i> ).....	39

## Lista de tabelas

Tabela 1. Fatores de Riscos.....	81
Tabela 2. Desafios em desenvolvimento tradicional e Àgil.....	93

# Acrónimos

GDD - Global Development and Delivery

QA - Quality Assurance

PM - Project Manager

VOD – Video On Demand

CMS - Content Management System



# Capítulo 1

---

## Introdução

A globalização tem gerado grande desafio para as empresas de Engenharia de Software. Atualmente os projetos têm sido desenvolvidos em ambientes geograficamente distantes, caracterizando o Global Development and Delivery (GDD). O GDD é caracterizado sempre que um ou mais recursos humanos são envolvidos no projeto fisicamente distantes. Pode-se dizer também que uma equipe global de desenvolvimento de *software* está muitas vezes em países diferentes, porém colaborando em um mesmo projeto de qualquer natureza (criação ou manutenção de *software*).

Alguns desafios são de grande relevância durante a implementação do GDD, como por exemplo a diferença de fuso horário onde as dificuldades de coordenação, sincronização e comunicação, principalmente durante a transferência de trabalho de um centro de desenvolvimento para outro, podem ser apontadas como consequências.

O objetivo deste trabalho é analisar mais detalhadamente porque é esse o caso e o que poderia ser feito para melhorar a situação na prática no trabalho cotidiano das empresas. Nesta tese, os desafios em GDD são discutidos com base em causas profundas. As soluções para esses desafios são analisadas de ponto de vista dos processos, pessoas, tecnologia e a apresentação das soluções para os mesmos.

Neste cenário, muitas vezes não é possível conduzir o desenvolvimento de *software* de maneira individual. Pessoas têm de trabalhar em equipas, o esforço tem de ser planeado, coordenado e acompanhado, bem como a qualidade do que se está produzindo tem de ser sistematicamente avaliada.

Este trabalho é uma combinação de um relatório de estágio com uma monografia sobre GDD com o que pretende-se uma análise profundo sistematizado um conjunto de características e de práticas correspondentes a este tipo de desenvolvimento.

Este trabalho está organizada da seguinte forma: este capítulo consiste na introdução ao tema e na definição dos objectivos deste trabalho; o Capítulo 2 está orientado para a visão geral do GDD, o processo de coordenação e colaboração entre as equipas que trabalham remotamente e apresentando os principais benefícios, o Capítulo 3 faz uma apresentação de um projeto desenvolvido na empresa GlobalSoftware onde fiz o estágio, o enquadramento do estágio, as duas funções que assumi na empresa, a descrição detalhada da organização em que me inseri e sublinhando um dos projetos em que participei – o GlobalSoftware; o Capítulo 4 está orientado para a listagem dos mitos e realidades na área da globalização e do desenvolvimento de *software* começando com uma abordagem dos mitos mais importantes, outros secundários, os mitos sobre os processos de modelagem e os vividos dentro do estágio. Finalmente no Capítulo 7 são apresentadas as considerações finais para este trabalho. A execução de aplicações numa rede distribuída globalmente fornece muitas vantagens, ao mesmo tempo, significa novos desafios. Este trabalho mostra que, embora o GDD seja comum, as empresas devem ponderar cuidadosamente os benefícios e custos de fazer o trabalho num ambiente distribuído.

A experiência prática mostra que a separação espacial e temporal das pessoas afeta profundamente o desenvolvimento de *software*, tornando mais difícil a sua realização devido a inúmeros motivos que variam desde a ausência de um idioma e uma faixa de

horário comum a até a falta de confiança e de senso de equipa entre as pessoas envolvidas.

## 1.1 Motivação

Todos os dias somos confrontados com a globalização tanto profissionalmente como nas nossas vidas privadas. A globalização é um termo que descreve o processo de transformação dos fenômenos locais em fenômenos globais. O termo é por vezes utilizado para descrever a ameaça ou a prosperidade. A prosperidade poderia ser visto como a globalização, aumento da acessibilidade de bens de consumo anteriormente apenas reservadas para os pouco privilegiados. A globalização também levanta questões morais sobre como agimos como uma sociedade global e particularmente na indústria de *software*. O *software* facilitou a globalização, fazendo a troca de uma ampla gama de serviços fáceis e rápidos através das distâncias geográficas. Na verdade, o *software* tem sido tão bem sucedido no sentido de facilitar essas trocas, uma vez que o serviço de desenvolvimento de software em si pode ser trocado entre lugares distantes. Em geral, essa indústria parece ter adotado uma visão positiva da globalização. Este acolhimento fez a deslocação de postos de trabalho de desenvolvimento de *software* cada vez mais comum. A deslocação de trabalho parcial deixa uma nova necessidade de colaboração entre distância geográfica. No entanto, é claro que os envolvidos nesta prática são e podem responder a essa mudança ou são só mitos?



## Capítulo 2

---

### GDD

O objetivo deste subponto é avaliar as recomendações e aumentar o conhecimento sobre o uso com base na literatura existente e resultados empíricos. Somos obrigados a diminuir o número de defeitos, aumentar a eficiência do trabalho e a oferecer as melhores oportunidades de aprendizagem.

A tese contribui para o entendimento de questões relacionadas com o desenvolvimento global de *software* por meio de projetos dispersos.

O crescimento da tecnologia tende a aumentar e as organizações de desenvolvimento terão de enfrentar novos desafios, porque são eles os responsáveis pela entrega do valor do negócio com uma força de trabalho globalmente diversificada.

O desenvolvimento global exigirá o domínio dos princípios de gestão de projetos do núcleo, tais como orçamentos e estimativas, otimização de recursos e de medição como também o acompanhamento da produtividade combinado com a consciência cultural.

Para ser bem-sucedido, devemos aprender a aplicar novas ferramentas, juntamente com as técnicas para assim envolver todos os membros da equipa de desenvolvimento.

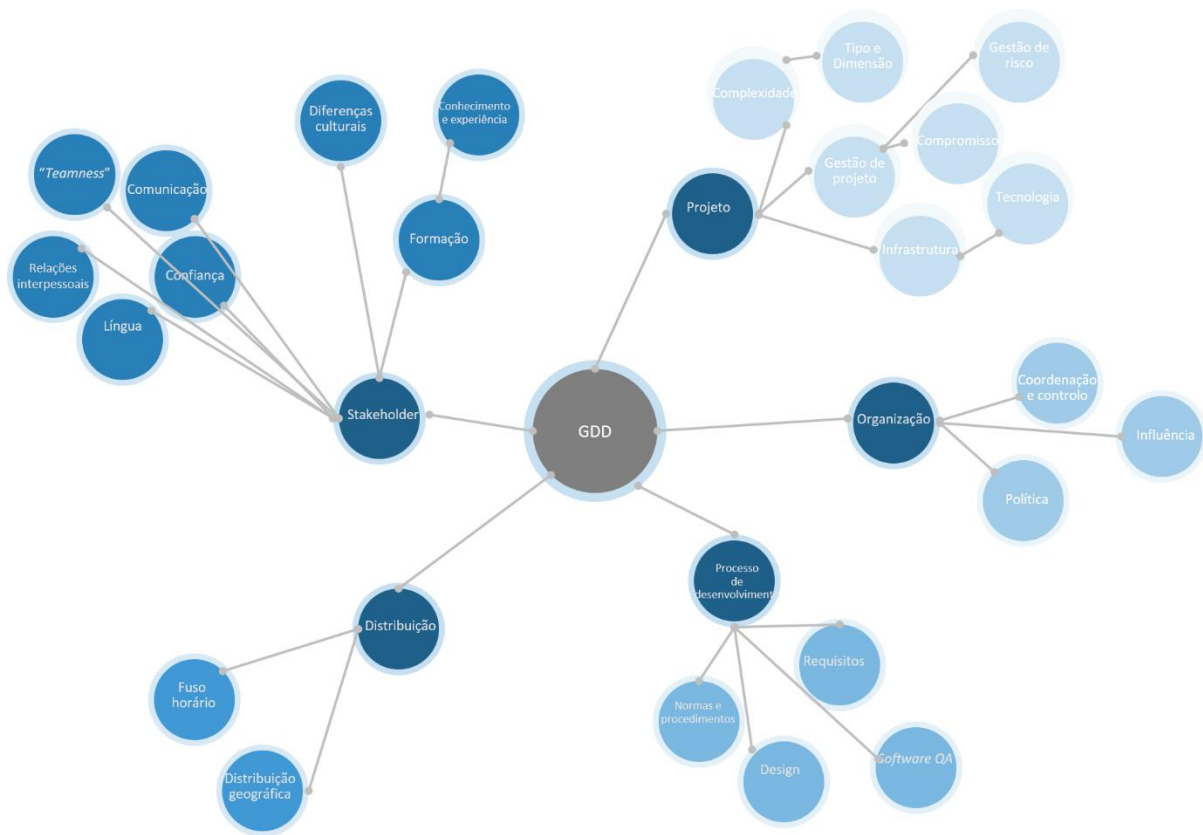


Figura 1. Mapa do desenvolvimento do projeto GDD

A base do GDD é constituída por 5 fatores: Organização, Stakeholder, Distribuição, Processo de Desenvolvimento e o Projeto (Figura 1). A dimensão de um projeto envolve um específico desenvolvimento de *software* centrado em coordenação geral de trabalho entre os colaboradores, comunicação, contacto com os clientes e solução dos conflitos. Os fatores apresentados no mapa facilitam a identificação de tópicos que conduzem à implementação de estratégias para minimizar ou antecipar os problemas.

A estratégia de negócios das empresas GDD consiste em estarem relacionadas com o fator custo, funções dos governos estrangeiros, diferenças em recursos e capacidade de supervisionar os concorrentes estrangeiros. O fator custo em particular e também o fator qualidade produzem uma vantagem comparativa em relação aos outros.

A preparação ou prontidão tecnológica também é mencionada como medida da agilidade com que a economia se adapta às tecnologias existentes e aumenta a produtividade dos

seus segmentos de mercado. O relatório afirma que a tecnologia tem-se tornado um importante elemento para que as organizações ampliem a sua competitividade e prosperem num ambiente globalizado. É nesse cenário que se desenvolveu o tema da tese e que contribui para o estudo do desenvolvimento global de *software*.

Dada a flexibilidade da organização, nem sempre as estratégias definidas são possíveis de se cumprir, por isso, devem ser providenciadas revisões periódicas para avaliar a aplicabilidade e adequabilidade da política em relação à organização.

Para atingir o sucesso, apesar de ter ganho em produtividade, requer muito esforço no sentido de colaboração, controle das atividades e qualquer decisão influencia no planeamento dos projetos. Além disso está a destacar-se a importância da distribuição no processo de desenvolvimento de *software*.

De acordo com o Herbsleb [1], a distância tem um impacto sobre as três principais formas de cooperação dentro de uma equipa: comunicação, coordenação e controlo. O desenvolvimento de *software* como uma equipa local é uma tarefa desafiadora, mas o desenvolvimento de software como uma equipa global de software é ainda mais desafiador, devido às distâncias.

## **2.1 Comunicação, Coordenação e Colaboração entre as equipas globalmente distribuídas**

A comunicação é a troca de informação entre os membros da equipa, seja formal ou informal, que ocorre na interação planeada ou improvisada [2]. Em equipas distribuídas, a falta de comunicação cara-a-cara surge como um dos principais obstáculos a uma cooperação remota.

A comunicação está representada no centro do modelo para enfatizar a sua importância e o facto de que ela permanece em todas as fases do processo (Figura 2). Sem uma comunicação eficaz, nenhuma abordagem à gestão do risco tem viabilidade. Embora a comunicação facilite a interação entre os elementos do modelo, existem igualmente níveis mais elevados de comunicação a considerar.

A comunicação permanece em todas as fases de um processo. Embora a comunicação facilite a interação entre os elementos, existem igualmente níveis mais elevados de

comunicação a considerar. Isso inclui níveis dentro do projeto de desenvolvimento, dentro da organização do cliente e muito especialmente através das fronteiras entre o utilizador e o cliente (caso este seja diferente do cliente).

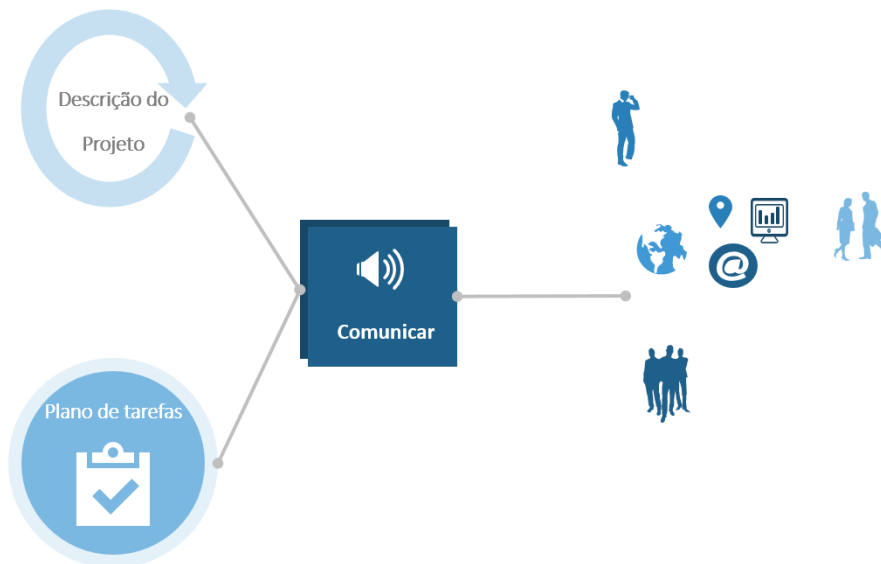


Figura 2. Comunicar no ambiente GDD

As equipas de sucesso são caracterizadas pela comunicação eficaz independentemente da comunicação virtual ou não entre todos os membros. Riscos, problemas e crises ocorrem dentro das organizações quando a comunicação é quebrada. A comunicação está subjacente ao sucesso da gestão e constitui os alicerces do funcionamento das equipas envolvidas no projeto – quer sejam equipas constituídas exclusivamente por elementos da organização que empreendem o projeto, quer sejam equipas distribuídas.

Por exemplo, o Herbsleb [1] relata que a comunicação no GDD é menos efetivo e então ele propõe trabalhar em processos compatíveis para reduzir a quantidade de comunicação necessária para alcançar mais comunicação efetiva. Isto está de acordo com outros pesquisadores que sugerem a superação de desafios GDD reduzindo a intensa colaboração [3], aumentando a documentação formal [4] e trabalhar em fatores organizacionais tais

como processos, estrutura e alinhamento das metas [5]. No entanto, o sucesso de processos ágeis em GDD projeta para outra direção, como os métodos ágeis dependem de uma estreita colaboração e comunicação informal frequente cara-a-cara ao invés de documentação longa. Alguns estudos sugerem que as práticas ágeis superam os desafios GDD: por exemplo, Holmstrom [6] especificou que as práticas ágeis são úteis para reduzir a comunicação, coordenação e problemas de controle, enquanto Layman [7] propõe que metodologias ágeis dependentes da comunicação informal podem ser usadas em projetos GDD, apesar de obstáculos geográficos, técnicos, temporais e linguísticas.

A coordenação é definida por Malone e Crowston [8] como "gestão de dependências entre as atividades." De acordo com esta teoria, a coordenação é obtida por mecanismos de um ou mais coordenações: cada um aborda um ou mais dependências. Por exemplo, no contexto de uma equipa ágil de *software*, as ferramentas, as atividades diárias como as reuniões, bem como a documentação, como o chefe de projeto, são todos considerados mecanismos de coordenação. Da mesma forma, no contexto de um projeto de GDD [5], um conjunto de mecanismos de coordenação de diversas naturezas é analisado, tais como a estrutura centralizada da equipa, documentação, ferramentas de comunicação e reuniões periódicas.

Além disso, definição de mecanismos de coordenação abrange conceitos tais como protocolos sociais e trabalho de articulação que parecem promissores para compreender não só a coordenação no GDD, mas também a diversidade das práticas coordenativas e de comunicação, a sua criação e evolução. Esta discussão terá continuação.

A distribuição global de desenvolvimento de *software* tem se tornado comum na última década. Ao nível tecnológico, as inovações em tecnologias de informação e comunicação, feitas pela eliminação da percepção de distância, tornam possível a cooperação em projetos de desenvolvimento de *software* em um modo distribuído, no entanto, atualmente há pouca pesquisa e experiência nesta área emergente. Os poucos estudos existentes [9] [10] relatam inúmeros problemas em projetos distribuídos. O tempo, a política, infraestrutura e cultura, associados com a dispersão geográfica do trabalho, torna mais difícil a gestão entre locais dependentes de trabalho, coordenar e controlar o trabalho distribuído. Além disso, os mecanismos de coordenação e de controle tradicionais são menos eficazes em projetos globais. No entanto pouco se sabe sobre o sucesso atual e metodologia suporte dentro de projetos distribuídos globalmente. Está a faltar conhecimento sobre como distribuir eficazmente o desenvolvimento de *software* em "padrões de colaboração" para usar as diversas atividades e

tarefas. Por isso, o meu interesse é explorar e melhorar a prática do desenvolvimento de sistemas de *software* em um ambiente distribuído globalmente.

A figura 3 identifica as áreas funcionais em que equipas distribuídas colaboram durante as várias etapas de um projeto. Essas áreas funcionais como a análise, o *design*, o desenvolvimento, a manutenção para questões de grupo, controlo, coordenação e padrões de colaboração e ferramentas identificados durante o planeamento de um projeto. Esse modelo une os diferentes elementos do plano, actividades de coordenação e de controlo relevantes.

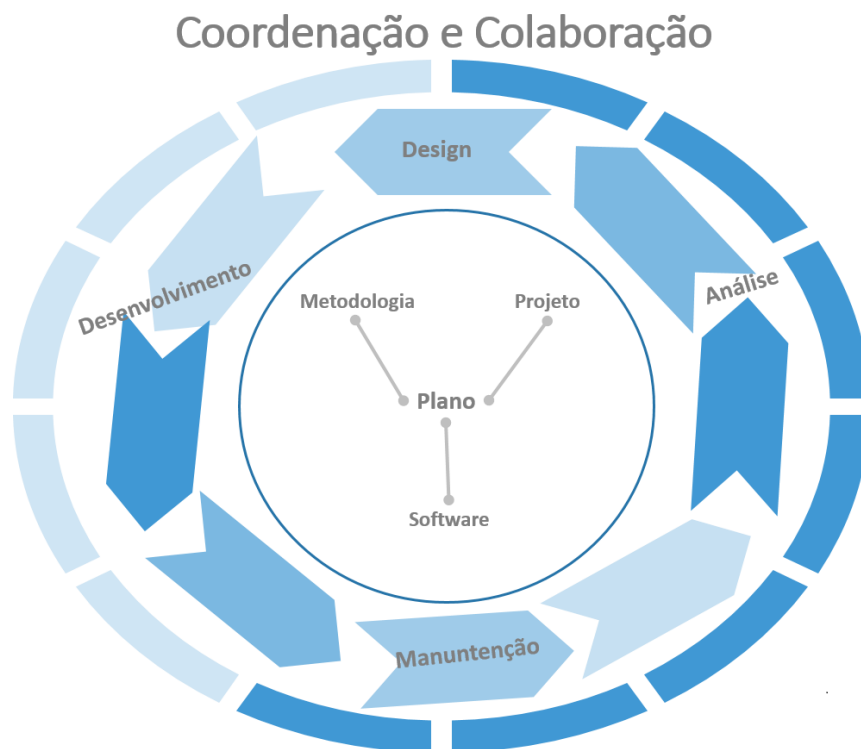


Figura 3. Coordenação e Colaboração (Kotlarsky 2002)

O objetivo da coordenação e colaboração é tomar decisões informadas e eficazes, em tempo útil, relativamente aos problemas que ocorrem no dia-a-dia e respetivos planos de mitigação. Isto constitui um passo do processo, o qual deverá ser facilmente integrado com as atividades de rotina associadas aos processos da tomada de decisão, estabelecidas no projeto.

A principal abordagem é o suporte (técnico, metodologia e social) necessário para a coordenação e a colaboração de equipas distribuídas no desenvolvimento de *software*.

Os vários fatores que influenciam as maneiras de fazer as equipas distribuídas colaborar são os fatores flexíveis e os fatores fixos; internos e externos. De seguida explicar como esses fatores influenciam as equipas e a maneira de colaborar: alguns dos limites / lacunas são a distância geográfica e as diferenças dos fusos horários - são fatores externos e fixos.

A distância é uma pré-condição, há sempre distância entre os locais. Quanto às diferenças de tempo, o horário de trabalho de sobreposição define quando é que as comunicações síncronas são possíveis. No caso da Austrália, Nova Iorque ou Índia e San Francisco não há sobreposição do horário de trabalho. As diferenças culturais são um fator muito importante que influencia a maneira de colaborar das pessoas de diferentes culturas. Estas são as diferenças de metodologias (ou processos seguidos), governança (diferenças na estrutura de gestão, regras, práticas implícitas) e as lacunas de infraestrutura, que incluem o acesso a serviços básicos como eletricidade e telecomunicações (capacidade), também diferentes (versões) de *hardware*, plataformas, e aplicações. Lacunas em infraestruturas técnicas e metodologias podem ser reduzidas, mesmo eliminadas, assim como as diferenças de governança também são possíveis de reduzir, apesar de ser bastante difícil.

O próximo fator é a tecnologia / Internet. Esta é um fator externo flexível porque, por um lado há uma série de ferramentas e tecnologias disponíveis no mercado e por outro lado, a disponibilidade de ferramentas e a escolha das pessoas envolvidas na colaboração remota.

Dentro da organização, temos uma distinção entre dois fatores (internos) que aplicam em diferentes níveis, estes organizacionais e individuais; fatores organizacionais e humanos, ambos os fatores flexíveis. Ao nível individual, temos visto que as pessoas que no mínimo numa reunião cara-a-cara, que ao mesmo tempo têm algum histórico de trabalho em conjunto colaboram melhor e com mais sucesso. Além disso, a motivação ao nível individual é muito importante para a colaboração bem-sucedida entre as equipas distribuídas. Metodologia, infraestrutura e governança podem ser consideradas como fatores organizacionais, mas também porque estas são as lacunas ao nível organizacional entre os locais envolvidos no mesmo projeto. Para cada projeto, estudámos como parte da coleta de dados empíricos, cobrimos todos os fatores acima mencionados, a fim de entender melhor os cenários. Além disso, estamos a acompanhar as variáveis tais como o tamanho da empresa e do projeto, o nível de experiência dos membros da equipa e outros fatores que podem influenciar o sucesso ou o fracasso de um projeto.

Durante as fases iniciais do projeto, as partes interessadas coordenam as convenções e criam redes de comunicação informal. Os obstáculos na comunicação informal (por exemplo, barreiras organizacionais) pode levar a mal-entendidos nas convenções e na lógica.

Há muitos benefícios potenciais que podem surgir de GDD: menores custos de desenvolvimento devido à poupança de salários, diminuição do tempo de desenvolvimento devido ao fuso horário, tempo reduzido para o mercado e acesso aos desenvolvedores mais talentosos [11]. No entanto, uma vez que as equipas GDD são espacialmente distribuídas, eles têm de lidar com distâncias temporais, geográficas e sócio-culturais [5], resultando em grandes dificuldades em coordenação e comunicação. Essencialmente, como Herbsleb [1] sugere, "o problema fundamental é que muitos dos mecanismos que funcionam para coordenar o trabalho num ambiente distribuído estão ausentes ou interrompidos num projeto distribuído."

## 2.2 Benefícios GDD

Apesar dos desafios bem conhecidos, o GDD também apresenta uma variedade de benefícios. Isto é, eles contribuem com as metas organizacionais de nível superior, tais como a redução de custos e o aumento da eficiência. Além disso existem benefícios que afetam mais diretamente as equipas e o desenvolvimento de *software* como processos e tarefas.

Atualmente há uma tendência pela tentativa de reduzir as distâncias temporais e sócio-culturais envolvidos. Esta abordagem assume que os benefícios de GDD não justificam plenamente o desenvolvimento de *software* verdadeiramente global, pelo contrário, nós encontramos equipas que mudam o horário de trabalho para aumentar a sobreposição de horas com colegas remotos, visando, assim, para um "dia virtual de 8 horas".

### 2.2.1 Redução de Custos

O benefício mais original e o mais procurado tem sido o de reduzir o custo de desenvolvimento [3]. A base para este benefício é que as empresas estão a globalizar as atividades de desenvolvimento de *software* para promover os funcionários localizados em economias *lowercost*. A diferença de salários entre as regiões pode ser significativa. As empresas estão a

olhar para os locais alternativos que oferecem taxas de atrito mais aceitáveis com a promessa continuada de trabalho mais barato.

### **2.2.2 O acesso a grandes forças de trabalho multiquificada**

O GDD oferece a possibilidade de utilizar grandes quantidades de mão-de-obra qualificada pela coordenação através da distância [12]. As empresas têm a oportunidade de expandir as atividades de desenvolvimento de *software* para incluir as contribuições de muitos trabalhadores qualificados, onde quer que esteja localizado, para formar corporações globais virtuais [4].

### **2.2.3 A redução *Time to Market***

Um benefício controverso tem sido o da abordagem “follow-the-sun”, descrito em pormenor por Carmel [13]. A eficácia do fuso horário é o grau em que a organização gere os recursos em vários fusos horários, maximizando a produtividade, aumentando o número de horas durante um dia de 24 horas em que o *software* está a ser desenvolvido pelas equipas. A eficácia de fuso horário é maximizada para abranger 24 horas do dia, isto é conseguido através da entrega fora do trabalho de uma equipa no final do seu dia a outra equipa localizado noutra fuso horário. A abordagem pode ajudar as organizações que estão sob uma forte pressão para melhorar o tempo de colocação no mercado [14].

### **2.2.4 A aproximação do mercado e dos clientes**

Ao estabelecer subsidiárias perto da localização dos clientes, o GDD permite-lhe desenvolver o *software* perto de clientes e aumentar o conhecimento do mercado local [4]. A criação de novos empregos pode criar boa vontade com os clientes locais, possivelmente resultando em mais contratos [15]. Na verdade, pode ser uma necessidade para os negócios estarem mais perto dos clientes a fim de expandir para outros mercados. Embora os benefícios que são bem conhecidos

tendem a afetar facilmente a política da empresa e por vezes, benefícios “escondidos”, que oferecem um grande potencial e efetivamente contribuem nas decisões estratégicas da empresa. Enquanto alguns dos benefícios que identificamos são aplicáveis na organização, eles parecem também afetar a coordenação e colaboração entre as equipas.

Se concentrarmos em aproveitar toda a gama de benefícios, defendemos que as empresas podem colher mais recompensas das atividades globais.

### **2.1.5 Inovação e partilha de práticas melhores**

As demandas do ambiente global de negócios estão sempre à espera do *software* inovador e de alta qualidade que atenda todas as necessidades [16]. As organizações podem aproveitar-se de uma maior inovação e melhores práticas partilhadas que surgem a partir da colaboração de membros da equipa que vêm de diferentes origens nacionais e organizacionais [10]. Em grandes organizações complexas, indivíduos independentes descentralizados interagem em forma de auto-organização para criar resultados inovadores e emergentes [16]. As empresas que se expandem para outros países, a fim de explorar o talento foram denominados de "buscadores de conhecimento" [17].

### **2.2.6 Alocação de Recursos**

Como uma extensão para o benefício de acesso a grandes grupos de trabalhadores multi-qualificados, a organização pode beneficiar do trabalho a baixo custo em outros países. Muitas vezes é mais estratégico ter as atividades ao mesmo tempo, evitando a turbulência do empregado, a reação associada a reduções da força de trabalho [18]. As mudanças na alocação podem aderir ao desafio de substituir pessoal isolado e em vez de criar tarefas de ampliação de habilidades e trabalho em equipa eficaz [10].

### **2.2.7 Melhorar a modularização das tarefas**

A importância de uma separação de interesses ao decompor o trabalho em módulos, é extremamente relevante para gerir a complexidade de coordenação. A natureza da GDD leva as equipas a dividir o trabalho em módulos independentes [19], sem interferências. Isso permite que as decisões sejam tomadas isoladamente sobre cada componente. As tarefas de

trabalho/funções/módulos, diminui as interdependências e, conseqüentemente, os custos de coordenação. Por exemplo, a ramificação do código-fonte permite o desenvolvimento de *software* trabalhem no código-fonte em paralelo, e fundindo as seções, uma vez que têm sido desenvolvidos [20].

## **2.2.8 Custos de Coordenação Baixos**

Embora reconheçamos que a distância temporal pode revelar-se um desafio para as equipas distribuídas, também pode ser visto como um benéfico em termos de coordenação: os custos de coordenação são reduzidos quando os membros da equipa não estão a trabalhar ao mesmo tempo [21]. Uma unidade de trabalho pode concluir o trabalho durante as horas de folga da pessoa que solicitou esse trabalho. Em essência, os custos de coordenação são reduzidos uma vez que nenhuma coordenação direta ocorre quando duas pessoas não estão a trabalhar ao mesmo tempo.

## **2.2.9 Autonomia das equipas**

Gumm [22] descobriu que a distribuição organizacional e geográfica de "unidades de desenvolvimento de *software*" implica um certo grau de autonomia para cada unidade. O estudo relatou que esta autonomia permitiu a necessidade de manter as diferentes culturas de cada equipa. Isto foi considerado necessário para manter a qualidade do trabalho mesmo se este por sua vez levasse à sincronização dos processos individuais.

## **2.2.10 Comunicação formal**

Desde que a comunicação assíncrona depende de tecnologias como e-mail e fax [23], a comunicação síncrona é geralmente deixada de lado [24]. Este prevê o aumento da rastreabilidade e responsabilidade [25] e permite a participação de diversas partes interessadas, independentemente da localização geográfica.

## **2.2.11 Documentação melhorada**

DeLone [26] afirma que as equipas distribuídas têm um maior enfoque na documentação a fim de auxiliar a sua comunicação. Gumm [22] relatou isso como uma vantagem, já que a documentação é melhor suportada dentro de configurações do projeto distribuídos. A informação é documentada e distribuída eletronicamente em vez de só falada cara-a-cara, o que permite a repercussão dos conhecimentos específicos do projeto em ambientes distribuídos.

### **2.2.12 Processos claramente definidos**

Independente da maturidade do processo de um projeto, a definição e estruturação de processos é um desafio [22]. As definições do processo são compiladas com mais atenção.

A maioria dos benefícios identificados aplicam-se pelo tempo e pelo processo/tarefa. Por exemplo as tentativas de reduzir os custos de coordenação de trabalho eficaz, enquanto ao mesmo tempo que desejam promover o GDD através da partilha de inovação e de boas práticas entre as equipas.

A comunidade GDD ainda tem de chegar a um consenso sobre quais dos benefícios são realistas e se são praticáveis. Por exemplo, ainda não está bem claro até que ponto a poupança imediata de custo pode ser atingida.

Mais provavelmente, certos benefícios só podem ser realistas em contextos específicos, enquanto alguns benefícios podem ser mutuamente exclusivos.

## Capítulo 3

---

### Projeto Desenvolvido

Este capítulo apresenta todos os aspetos de um projeto de *software* já desenvolvido. O objetivo é conhecer o projeto e, mais especificamente, as práticas utilizadas na coordenação do trabalho sobre as responsabilidades de equipas distantes entre si e sob diferentes aspectos, porém, com propósito e objetivos compartilhados. Também considere importante investigar como um determinado problema está sendo tratado ou solucionado pelos membros da equipa.

Inicialmente a minha ferramenta era o conhecimento acumulado na área e as novas evidências que foram estabelecidas a partir da observação.

A GlobalSoftware cria aplicações, uma linguagem de *design* para aplicações, onde o conteúdo tem o palco principal. O projeto SoftwareX implicou um total de 9 recursos humanos, com apenas 4 na central e 5 no local distribuído. A GlobalSoftware dispôs de um chefe de projeto, um técnico de informática, *designer*, testador, programador para mobile e Tablet android, programador para mobile e Tablet iOS, programador para Smart TV, programador para Apple TV e um programador para Roku box.

Posteriormente e em consequência de ações internas, a GlobalSoftware criou uma aplicação - projeto SoftwareX para interfaces tangíveis, aliadas muito significativas nessa empreitada, proporcionando ao usuário uma interação mais natural com os dispositivos e plataformas.

O projeto passou por todas as fases do ciclo de vida de um *software* começando com a especificação de requisitos até a análise do teste. Fez uso de um padrão hierárquico e outros padrões de navegação, que faz com que a aplicação seja rápida, fluída e fácil de utilizar, permitindo que os utilizadores localizem rapidamente o que precisam.

Dentro das características desse tipo de projeto está o que a maior parte do trabalho foi realizado à distância e por recursos humanos especialistas, alocados em fornecedores fora do país que empreenderam o projeto. Foi muito comum trabalhar com pessoas de outras origens, costumes e outra língua natal, porém, regularmente com fluência no idioma inglês. Nesse contexto, a infraestrutura de comunicação via satélite e por fibras ópticas que permitiu a disseminação do uso da internet, possibilitou também a configuração de escritórios virtuais globais possibilitando a integrantes das equipas de projeto estarem num lugar-comum.

A metodologia utilizada foram as metodologias ágeis trabalhando com um *feedback* constante, o que permite a rápida adaptação a eventuais mudanças nos requisitos. A comunicação foi feita através de e-mail e reuniões semanais entre o chefe de projeto e o cliente. O volume e a frequência das mudanças foram muito ricas o que exigiu a constante interação com o cliente. Um outro ponto positivo das metodologias ágeis são as entregas constantes de partes operacionais do *software*. Desta forma, o cliente não precisa esperar muito para ver o *software* funcionando e avaliar acerca do produto solicitado.

A integração e o teste contínuam a possibilitar a melhoria na qualidade do *software*. Não é assim necessário adicionar uma fase de integração de módulos, uma vez que eles são continuamente integrados e os eventuais problemas são resolvidos constantemente.

Durante o estágio coletei a perspectiva da empresa GlobalSoftware participando em projetos globais. Os critérios foram os processos de desenvolvimento, especificação e integração. Tive acesso a documentação e registros - documentos digitais e registros que fornecem informações sobre o projeto e locais envolvidos. Uma outra ferramenta foi a observação direta, participação em reuniões e outras comunicações entre os *sites*, tanto quanto possível.

Embora os principais meios de comunicação usados são sistemas de e-mail, telefone e vídeo conferência, também fazemos uso do Software Social, uma gama de sistemas de software que

nos permitem interagir e compartilhar informações, tais como: Skype, web para reportar os defeitos/*bugs* (Mantis), a ferramenta de gestão de projeto e colaboração baseada em web (Basecamp) e o OneDrive para o Business que é parte integral do Office 365, que oferece um local no *cloud* onde podemos armazenar, compartilhar e sincronizar os arquivos de trabalho, atualizar e compartilhar os arquivos de qualquer local permitindo até trabalhar em documentos do Office simultaneamente com os outros.

Com o uso dessas ferramentas, explorando a constituição, estabelecimento e manutenção de práticas comunicativas e coordenativas em GDD, é possível construir e manter relações sociais entre os colegas de trabalho distribuídos para facilitar a partilha de conhecimento dentro da organização, para apoiar a gestão do conhecimento e para encorajar a comunicação informal entre os membros da equipa distribuídos e os clientes internacionais.

A combinação das variáveis presentes no projeto desenvolvimento e o estudo das pesquisas já realizadas sobre o tema do desenvolvimento global de software, indicam a existência de oportunidades a investigar. A coordenação do trabalho nessa categoria de projetos é uma dessas oportunidades.

## 3.1 Contexto do estágio

Inicialmente, a abordagem metodológica escolhida neste relatório foi um estudo teórico e um estudo de caso. Depois de conhecer a organização e entrevistar os membros, surgiu logo a oportunidade de um estágio.

O estágio começou e com ele a definição dos objetivos profissionais. Analisei as características do estágio, os objetivos e finalidades que este perseguia e possibilitaram-me a conceção de prioridades.

O que me chamou à atenção foi essa específica empresa que atende de forma eficaz e eficiente as demandas, acompanhando as mudanças e transformações do mercado associado à era da tecnologia e da informação. As inovações tecnológicas promoveram o processo de globalização.

O estudo do projeto tem sido facilitado pela GlobalSoftware (GlobalSoftware que é um pseudônimo), uma empresa de fornecimento de serviços de tecnologia e globalização. Todos os detalhes neste relatório são anónimos, devidos às informações confidenciais de negócios. A empresa GlobalSoftware é um ator global no fornecimento de soluções tecnológicas e globalização de serviços a uma ampla gama de clientes em todo o mundo. GlobalSoftware tem a sua base de clientes em vários grandes grupos da indústria, tais como meteorologia, comunicações, alta tecnologia, entretenimento etc. A maioria de seus grandes clientes são clientes de longo prazo. Também possui dois centros de distribuição em Portugal, Espanha e na Inglaterra (a central) com dúzias de funcionários.

Embora a entrada da GlobalSoftware na globalização seja relativamente jovem, já desenvolveu formas padronizadas de lidar com questões comunicacionais, de organização e de infraestrutura quando se trata de desenvolvimento e manutenção de *software* distribuídos globalmente e manutenção. GlobalSoftware leva a vantagem de usar esses padrões, que são criados através de experiência com as melhores práticas da indústria. Como uma interna, posso confirmar que estes métodos padronizados são úteis e faz com que o desenvolvimento global de *software* apresentem mais vantagens do que sem eles, mas ao mesmo tempo também revela-se que

existem problemas causados pelas diferentes culturas. Embora os métodos padronizados possam trabalhar (verificadas por mim mesma) muito bem para os americanos e outros europeus, existem algumas questões que surgem, mais especificamente para os orientais, isto é, a barreira da língua.

A empresa GlobalSoftware apresenta a solução que ganhou um *Award*, a ferramenta que consegue alcançar mais de 100 plataformas e distribuí-las globalmente resultando em menos custos de desenvolvimento só com a ajuda de um simples clique. A ferramenta acelera o processo gerando uma aplicação para múltiplas plataformas.

A GlobalSoftware disponibiliza um alcance sem paralelo numa vasta gama de dispositivos, desde mobile, Tablet, Smart TV, Roku, Apple TV e consolas de jogos, oferecendo aos publicadores de conteúdos novas formas de se ligarem, cativarem e manterem próximos a sua audiência. Eles fornecem aos publicadores de conteúdos excelentes oportunidades para distribuírem, promoverem e venderem o seu conteúdo.

Na fase inicial do estágio assumi funções de *Quality Assurance*, efetuei uma retrospectiva do meu percurso, retirando ilações que mais tarde associei às expectativas iniciais em confronto com o encontrado. Este cargo é aplicável para as funções de garantia de qualidade em casos de desenvolvimento de *software*. A função essencial era garantir a qualidade para caucionar o atendimento dos requisitos das aplicações. Um esforço constante era melhorar as práticas de qualidade na GlobalSoftware para atingir os objetivos e entregar o produto com menos defeitos possíveis. Usava as práticas de testes persistentes por meio da otimização dos processos de produção existentes.

Dois meses depois passei a assumir as funções de Chefe de Projeto. Diariamente, o ciclo da realização, reflexão e reformulação surge como parte das tarefas diárias, no sentido de alcançar uma adequação do processo de gestão. Eu combinei técnicas de gestão de projetos com uma compreensão de desenvolvimento de *software* e entrega consistentemente dos projetos dentro do prazo, dentro do orçamento e com a qualidade garantida num ambiente Ágile.

## 3.2 Desafios em gestão de projetos distribuídos

As características de trabalho em equipas virtuais, como a dispersão geográfica, a dependência eletrónica, dinamismo estrutural e diversidade nacional, muitas vezes dificultam a inovação [27]. Esta observação aplica-se especificamente as equipas virtuais em desenvolvimento de *software* e implica vários desafios de gestão [28]. Os desafios da gestão de projetos distribuídos surgem porque a tarefa do projeto é dividida e distribuída em vários locais. A divisão e a distribuição da tarefa pode tornar-se mais difícil para os participantes do projeto para entender a tarefa, o seu objectivo [29] [30] e a sua própria contribuição para a tarefa global [10] [31]. A consideração de interfaces, a influência do subsistema e a carga de trabalho é, portanto, fundamental ao segmentar a tarefa em projetos de *software* distribuídos. Um aumento de acoplamento entre os segmentos de tarefas pode aumentar a necessidade de comunicação entre os locais, coordenação e integração e pode, assim, levar a um aumento do número de falhas [3] [32]. Além disso, quando os programadores de diferentes partes do mundo colaboram, a compatibilidade das ferramentas é um desafio comum. Cada região/país utiliza diferentes linguagens de programação, ferramentas de suporte, sistemas operacionais e ferramentas de desenvolvimento [33] [34] [32]. A comunicação é um desafio crítico na gestão de projetos de *software* [35]. Num projeto distribuído, a comunicação é ainda mais difícil quando os participantes são distribuídos através do tempo e espaço. A interação social é muitas vezes impedida pela troca de informações baixa, ausência de comunicação informal [14] e a falta de interação cara-a-cara [33] [36]. Essa confiança pode impactar negativamente [34] [32] a qualidade das decisões [4] [34], a criatividade [34] e gestão [37]. Além disso, as limitações de comunicação podem reduzir a visão geral do projeto e dos participantes e levar a erros e mal-entendidos [14]. Quando há falta de interação, as pistas verbais e não-verbais são limitadas. Isto pode incluir problemas, tais como sequências confusas de mensagens e perda de partilha de informação contextual [34]. Os participantes dos projetos distribuídas não compartilham sempre a mesma linguagem ou cultura organizacional. As barreiras linguísticas estão tipicamente presentes em projetos transnacionais quando os participantes não compartilham uma língua nativa comum [38] [36] ou normas de comunicação [34]. Essas barreiras podem resultar em erros de interpretação e informação não transmitida [33] [36] [32]. As diferenças na cultura de trabalho também podem dificultar um projeto distribuído. Estes obstáculos podem aparecer devido a distância em termos de comportamento da equipa ou da cultura organizacional [3] [33]. As dificuldades podem ser baseadas em diferenças entre os locais, em

equilibrar o coletivismo e o individualismo, na percepção da autoridade e da hierarquia [14] [29], e planeamento e pontualidade [4]. Isso pode levar à diminuição da capacidade de manuseamento de conflito, menor eficiência, ou mesmo paralisar um projeto de *software* distribuído [33] [39]. Em geral, quando os projetos estão distribuídos ao longo do tempo, espaço e cultura, é difícil obter o mesmo nível de coesão do grupo em equipas distribuídas [30]. Os participantes do projeto são menos propensos a se comprometerem com a organização do projeto e da sua missão, quando as diferenças culturais e a falta de interação cara-a-cara torna difícil estabelecer uma identidade clara do projeto [40] [31] [39]. Isso enfraquece a sinergia do grupo [40], aumenta o risco de conflitos [39] e pode diminuir a eficiência na fase inicial do projeto. Os projetos de *software* distribuídos são frequentemente caracterizadas por estruturas organizacionais horizontais [41]. Os processos, os métodos de desenvolvimento e a ênfase no envolvimento do usuário, possivelmente resultam em incompatibilidade [42] [43] [30]. No entanto, muitos dos desafios são muito persistentes e ainda precisam de ser abordadas.

A literatura existente tem promovido os benefícios inclusivamente da comunicação cara-a-cara, mas pouco se sabe sobre as implicações da comunicação e *feedback* no contexto do desenvolvimento.

Em conclusão, as organizações envolvidas na GDD precisam de se concentrar em um conjunto de processos que auxilia a organização da equipa e a distribuição do trabalho. Estes processos podem ter um tema técnico, tais como a integração, a ramificação, assim como temas sobre a distribuição de trabalho e estrutura de comunicação. Tendo estes processos de engenharia de *software*, organização da equipa e a comunicação por si só não são suficientes. Além disso, é necessário comunicarem de forma consistente [39] para assegurar que sejam entendidos e seguidos.

O GDD está a se tornar uma prática comum no desenvolvimento de *software*. É tanto proporcionar oportunidades interessantes e como apresentar desafios significativos para o sucesso dos projetos distribuídos.

Os nossos dados empíricos, no entanto, demonstram que há espaço para melhorias, mesmo num ambiente global.

### 3.2.1 Assumir funções QA

Um Software, desde que inicia a sua concepção até à sua disponibilização ao público-alvo, passa por uma série de processos. Um dos processos mais importantes é a execução de Testes de Software, na medida em que se procura a garantia de qualidade da mesma. A presença dos defeitos num processo de desenvolvimento ou implementação de um *software* irá comprometer a qualidade do mesmo. Por isso, é essencial identificar esses defeitos para que eles possam ser corrigidos. A correção dos defeitos é também constituída através da realização de Testes de Software. Devido à sua complexidade e importância, estes testes podem constituir uma parcela bastante relevante dos custos de um projeto. Quanto mais cedo um defeito for encontrado, menores são os custos associados à sua correção. Neste sentido, é essencial sensibilizar as organizações para a sua realização por profissionais, de modo a que o resultado final do projeto seja um sucesso. A atividade de teste é uma das etapas do ciclo de desenvolvimento de *software* e tem o objetivo de relatar possíveis defeitos existentes no sistema para que estes sejam solucionados. Nesta fase verifica-se se o comportamento do sistema está de acordo com o especificado nos requisitos levantados junto ao cliente. A partir desta atividade pode-se diagnosticar o grau de qualidade do sistema. O principal objetivo da realização do teste de *software* é reduzir a probabilidade de ocorrência de erros quando o sistema estiver em produção [8]. Para Pressman [9], "o teste de *software* é um elemento crítico da garantia de qualidade de software e representa a revisão final da especificação, projeto e geração de código". Por ser a última etapa, antes da entrega ao cliente, a fase de teste tem a responsabilidade de encontrar as falhas inseridas no decorrer do projeto.

Durante o ciclo de vida do desenvolvimento de software, a fase de testes raramente é um processo demorado. No entanto, diagnosticar e corrigir erros detetados no decorrer dos mesmos é um processo demorado, mas também muito produtivo.

Os programadores são responsáveis por componentes específicas ou áreas que lhes são atribuídas, mas os testadores têm de compreender o funcionamento global do *software*, quais as suas dependências e quais os impactos de um módulo sobre outros.

A combinação de todos esses componentes e suas melhores práticas é o que chamamos de *Quality Assurance*, e que por sua vez todo esse trabalho é realizado por pessoas, garantindo a qualidade de *software* do produto final entregue ao cliente ou ao usuário final.

Antes de iniciar as atividades, participei em reuniões virtuais via Skype com um QA da central a fim de me informar/dar-me dicas sobre o processo de teste da aplicação em particular e atenção dirigida para as tarefas a serem concluídas. Uma das minhas funções era controlar a qualidade e garantir a qualidade do trabalho das equipas de desenvolvimento, deixando comentários construtivos que abordassem os pontos fortes e os pontos fracos do trabalho a serem tomados em conta.

Um dos objetivos iniciais foram: tornar-me familiar com a empresa, equipa local e central e com os projetos que me foram atribuídos para estudar/analisar/testar, tal como explorar as ferramentas necessárias.

A qualidade era atingida através do uso das ferramentas certas e adequadas para a especificação, desenho, codificação e testes. A outra dimensão era trabalhar em equipa, comunicar com a equipa, pensar como uma equipa e usar uma linguagem comum que nos possibilitasse partilhar, explorar, experimentar, alterar e testar. Tudo o que é considerado um teste, é uma forma de controlo de qualidade. Verificar se um desenho cobre todos os requisitos da especificação, correr uma verificação total, medir o tempo de resposta, são todas atividades do controlo de qualidade.

É necessário levar a cabo as verificações de controlo da qualidade em cada etapa do desenvolvimento, idealmente em todas as entregas intermediárias- especificação de requisitos, análise, desenho, código, manuais do utilizador e documentação. Tudo, em suma, necessitamos definir atributos da qualidade para cada software.

A metodologia de teste usada é o de caixa-preta que se baseia nos requisitos funcionais do *software*. Como não tinha conhecimento sobre a operação interna do programa, o teste concentrava-se nas funções que o produto deve desempenhar. A partir da especificação são determinadas as saídas esperadas para certos conjuntos de entrada de dados.

Uma tarefa começava com a revisão de requisitos e desenho de casos de teste. As responsabilidades também incluíam a deteção de *bugs* da aplicação e seguidamente esses *bugs* eram atribuídos ao programador em causa para se efetuarem as correções.

Os defeitos particularmente encontrados eram funcionalidades implementadas incorretas ou omitidas, erros de interface, erros de comportamento ou desempenho e erros de iniciação e finalização.

Essas atividades são focadas no início das fases no ciclo de vida do desenvolvimento de *software* até à entrega final.

Uma outra responsabilidade requerida era criar um plano de garantia da qualidade. Cada vez que se executava um teste (uma vez por entrega *do milestone/drop*), verificava-se a qualidade da entrega e se correspondia ao plano acordado.

Como tinha referido antes, eu estava a desempenhar as funções de QA e também fazia uso das ferramentas Skype, Mantis, Basecamp e OneDrive com acesso total aos projetos internos e externos.

A equipa de QA também trabalha em conjunto com os Chefes de Projetos na criação de documentos e instruções e oferecem suporte e *feedback* aos colaboradores externos (sejam clientes ou pessoas suporte das lojas de Smart TVs online) e com os programadores a dar suporte para localizar os *bugs* via Mantis ou na hora do teste.

Do ponto de vista crítico, ao assumir essa função, tinha os pontos altos que eram calcular e melhorar a qualidade do *software*, entrar em contacto com os QAs das lojas Smart TV, partilhar experiências e ajudar a equipa a progredir. As abordagens finais baseavam-se em resultados.

### **3.2.2 Assumir funções PM**

Hoje em dia os projetos exigem uma integração muito apertada, obriga ao uso de formas mais criativas de desenho, de construção, teste e implementação de *software*. Um gestor não pode criar só um calendário de um projeto apenas através do proeenchimento de um ou dois *templates*. Uma forma de alcançar um resultado eficiente é adoptar metodologias mais “leves”, mais ágeis (no caso da empresa GlobaSoftwareX, usam a metodologia Scrum).

Construir *software* é uma tarefa complexa e complica-se ainda mais sobretudo se as pessoas trabalham em locais distantes.

Os projetos virtuais levantam questões delicadas que não têm a ver só com a natureza técnica. Estes projetos envolvem a delicada tarefa de gerir diferentes expectativas culturais, atribuir prioridades, estabelecer um sistema de controlo e definir responsabilidades por processos. A

estas tarefas crescem alterações subtis nos habituais problemas com custos, prazos e qualidade, enfrentados por todos os chefes de projeto que tentam entregar o *software* a um cliente.

A minha função era enfrentar os desafios diários e ter os clientes, equipas e projetos na mesma balança indiferente ao peso de cada parte. Assumindo as funções de Chefe de Projeto, era responsável pelo planeamento e execução de projetos para clientes internacionais num ambiente Agile. Eu combinava as técnicas de gestão de projetos com uma compreensão de desenvolvimento de *software* (aplicações móveis, aplicações Smart TV) e experiência do usuário para os entregar consistentemente no prazo, dentro do orçamento e com a qualidade requerida. Fazia revisões diárias dos planos de projeto e cronogramas; trabalho com os *designers*, programadores e equipe de *QA*, e continuamente ficar em contato com os clientes. Uma outra responsabilidade era melhorar o processo de desenvolvimento por meio da criação de documentação, identificação de lacunas, definição clara das funções de cada membro da equipa.

Eu assumi a responsabilidade de prestar atenção desde o início do estágio à forma de gerir os projetos com a máxima alerta e a máxima concentração. No início, a comunicação não era muito abrangente com o cliente, pois não tinha conhecimentos suficientes, com o tempo e mais experiência, conseguia arranjar soluções elegantes para os problemas que iam surgindo.

Sempre que iniciava uma actividade/projeto, criava um plano que definia os processos e tarefas a serem realizadas, as pessoas que iriam fazer o trabalho, os mecanismos para avaliar os riscos, controlar as alterações e avaliar a qualidade. Nunca estava completamente segura se os planos estavam corretos ou se era a pessoa certa a completar a tarefa até termos entregue o *software* com a qualidade esperada e nos prazos inicialmente definido e assim como dentro do orçamento. Uma outra função assumida era encorajar/motivar os programadores a trabalhar em conjunto como uma equipa, na ajuda recíproca, concentração no cliente e na qualidade do produto.

Estas funções assumidas fizeram-me crescer e evoluir consideravelmente ao longo dos meses de estágio, integrando as funções e responsabilidades no domínio da gestão de projetos.

## 3.3 A Organização

Um projeto conjunto de desenvolvimento de *software*, seja com uma outra empresa ou não, deve ser adequadamente gerido: as pessoas devem estar motivadas, os recursos corretamente atribuídos e a qualidade do produto devidamente preservada. Ou seja os objetivos são exatamente os mesmos da situação em que se tem toda a equipa num único local - o que muda completamente é a forma como se atingem esses objetivos.

A manutenção, a responsabilidade e a referência à documentação, bem como a capacidade de comunicação, são questões críticas em grandes equipas como elementos geograficamente dispersos.

As primeiras dúvidas eram: “O que fazer, que atitude tomar, que postura assumir?”. Essas dúvidas emergentes foram tratadas logo com o orientador da tese, o professor Fermé, de forma precisa e suficiente para solucionar o problema em questão. Uma das respostas para cada problema era ser flexível, dinâmica e sábio o suficiente para lidar com situações novas e inusitadas a todo o momento.

Para criar um grupo amigável, requeri a atividades como reuniões matinais “5 minutos”; motivar; ouvir; pedir opiniões/sugestões e criar um ambiente calmo e acolhedor.

### 3.3.1 Fatores Relevantes na Divisão do Esforço do

#### Desenvolvimento

Na maioria dos casos, a divisão do esforço foi relacionado com a capacidade tecnológica. Uma outra abordagem foi dividir o esforço de acordo com as fases do processo de desenvolvimento. A GlobalSoftware possui as seguintes especialidades: desenho de *software*, integração e testes, apoio ao cliente, etc – que servem como base para atribuição ao longo processo de desenvolvimento.

Com os prazos apertados, os recursos humanos constituíram a base principal para a divisão do esforço. A vantagem desde método foi reduzir os riscos do prazo, pelo facto de as tarefas maiores poderem estar atribuídas à parte que tem mais recursos. O esforço também foi distribuído de acordo com as ferramentas e os recursos financeiros. As ferramentas e o capital

forneceram uma alavanca contra o risco, porque podiam aumentar a produtividade, a qualidade e a capacidade de enfrentar os desafios inesperados.

### **3.3.2 A Gestão de Vários Componentes**

O maior desafio para este projeto foi a capacidade de se juntarem eficazmente as suas variadas peças, dispersas por locais remotos. A natureza de um projeto virtual coloca desafios especiais na gestão do risco e das diferenças culturais, na sua organização, na gestão da configuração do *software* e no processo de decisão.

Os riscos centraram-se em torno de questões apenas parcialmente visíveis ou questões que nem sempre são comunicadas em tempo oportuno. Todos os membros da equipa estavam conscientes das tarefas a desempenhar e das eventuais limitações associadas.

Para a manter a equipa forte, e ultrapassar quaisquer problemas técnicos, administrativos ou culturais, tive que manter a estrutura simples e comunicá-la bem.

O controlo das versões foi o exercício de documentar e guardar alterações. O projeto SoftwareX usou um sistema de controlo de versões que permitiram aos múltiplos utilizadores monitorizarem os próprios ficheiros históricos, bem como a ligação de versões de ficheiros individuais e uma configuração de projetos.

### **3.3.3 A Integração**

A integração tornou-se mais difícil por causa das várias versões de *software*. Como em muitas outras atividades do projeto, o planeamento da integração começou muito antes do início da própria atividade e envolveu o desenvolvimento de uma estratégia de integração, a aquisição das ferramentas adequadas, a escrita ou o uso de séries de testes, a determinação dos critérios de aceitação, a criação da documentação e o fornecimento do adequado nível de suporte.

Se não ia ser adequadamente planeada, a integração no projeto global virtual redundava numa miríade de módulos de *software* – cada um com um dado nível de inteireza, em termos de testes e documentação. Para conseguir obter bons resultados, foi preciso tornar-me um planeador, um organizador eficaz e criar um ambiente em que a equipa poderia ter um desempenho ao nível das capacidades de cada um. Um dos objetivos era estabelecer boas relações de trabalho com a equipa e representá-la de forma eficaz.

A experiência foi definitivamente positiva e encorajou a conduzir o desenvolvimento do projeto em paralelo.

### 3.3.4 O projeto SoftwareX

A empresa que concebeu o projeto SoftwareX, é um reprodutor de áudio e vídeo para reproduzir e organizar música digital, arquivos de vídeo e para a compra de arquivos de media digital. Os usuários podem comprar arquivos de media digital dentro da própria aplicação. A empresa vem ganhando e mantendo uma boa reputação ao mesmo tempo permite que os usuários organizem precisamente as músicas.

A aplicação explora mais de 40 milhões de faixas, permite a descoberta de artistas e faixas novas como também funciona em todas as plataformas Smart TV, sem limites de escuta.

Os requisitos são os sinónimos de funções, ou seja, tudo que o *software* deveria fazer. No entanto, os requisitos de software são, além de funções, objetivos, propriedades, restrições que o sistema deve possuir para satisfazer contratos, padrões ou especificações de acordo com o(s) usuário(s). De forma mais geral um requisito é uma condição necessária para satisfazer um objetivo. Portanto, um requisito é um aspecto que o sistema proposto deve fazer ou uma restrição no desenvolvimento do sistema.

Nesse caso, apresento de seguida os requisitos para o *Login* (requisitos mandados pelo cliente) – Realizar o *login*, a execução do *login* no sistema, o usuário deve fornecer seu *login* e senha. O *login* ao ser efetuado com sucesso, o usuário terá mais opções referentes aos módulos mencionados na descrição do sistema – Sistema possui apenas um tipo de usuário.

## Requisitos Login:

- 1. Descobrir
  - Não há necessidade de fazer o *Login*.
  - Pressionar o botão Descobrir para ter acesso a aplicação.
  - O *user* pode ouvir música durante 30 segundos.
  - O *user* tem acesso total ao *Mixes*.
- 2. *Free user*
  - *Login*
  - Se o *user* não tem acesso à conta, pode-se registrar na aplicação ou via *website*. A *password* é automaticamente gerada e enviada pelo email. A *password* pode ser alterada só no *website*.
  - O *user* pode ouvir música durante 30 segundos.
  - O *user* tem acesso total ao *music & flow*.
- 3. *Free trial*
  - *Login*
  - Se o *user* não tem acesso à conta, pode-se registrar na aplicação ou via *website*. A *password* é automaticamente gerada e enviada pelo email. A *password* pode ser alterada só no *website*.
  - Um *pop-up* com o tempo restante aparece. O *user* se pressionar o botão OK button, pode aceder à aplicação.
  - O *user* tem acesso total na aplicação.
- 4. *Premium user*
  - *Login*
  - Se o *user* não tem acesso à conta, pode-se registrar na aplicação ou via *website*. A *password* é automaticamente gerada e enviada pelo email. A *password* pode ser alterada só no *website*.
  - O *user* tem acesso total na aplicação.
- 5. *Premium + family*
  - *Login*
  - Se o *user* não tem acesso à conta, ele pode registrar na aplicação ou via *website*. A *password* é automaticamente gerada e enviada pelo email.
  - O *user* tem acesso total na aplicação e a possibilidade de adicionar vários *profis*.

O projeto foi implementado nas seguintes plataformas Smart TV representadas na figura abaixo:

Device	TV	AV	STB	2012	2013	2014	2015
Samsung	✓	✓		✓	✓	✓	✓
Samsung Tizen	✓	✓					✓
LG (WebOS, android TV, STA, Netcast)	✓	✓		✓	✓	✓	✓
NetTV/TP-Vision( Philips, Sharp, B&O)	✓	✓		✓	✓	✓	✓
Opera TV (Sony Bravia, Tivo, TCL, Hisense, Humax)	✓	✓	✓	✓	✓	✓	✓
Panasonic	✓	✓		✓	✓	✓	✓
Android TV (Vizio, LG, Sony, Philips, Hisense)			✓	✓	✓	✓	✓
Toshiba	✓	✓		✓	✓	✓	✓
Smart Alliance (Philips, LG, Panasonic, Toshiba)	✓	✓		✓	✓	✓	✓
Amazon Fire TV			✓			✓	✓
Sony Bravia	✓	✓		✓	✓	✓	✓
Vestel (38 other sub brands including STBs)	✓	✓		✓	✓	✓	✓
Foxxum (13 other sub brands)	✓	✓		✓	✓	✓	✓
NetRange(42 other sub brands)	✓	✓	✓	✓	✓	✓	✓
Canal +			✓			✓	✓

Figura 4. Plataformas Smart TV

### Outros projetos no mercado

O mercado do *software* é considerado um dos mais ativos do mundo entre o desenvolvimento e venda de produtos, porém existem obstáculos.

No entanto, desenvolver sistemas de *software* não é fácil, devido, entre outros aspectos, à consideração das necessidades reais de cada cliente, ao grande número de fases que constam nos processos de negócio, e às regras de negócios complexas e específicas de cada empresa. A fórmula do sucesso implica adicionar sempre a imagem a maturidade dos processos de desenvolvimento. A GlobalSoftware desenvolveu ao longo dos 5 anos projetos de entretenimento muito semelhantes aos encontrados no mercado do dia-a-dia. Os projetos que pesam mais são os projetos e manutenção, os demandados por incidências, mudanças no ambiente externo, novas necessidades e melhorias de manutenção.

A trajetória para ganhar os projetos no mercado de *software*, é a criação de um produto inovador, com alta qualidade, que atenda os requisitos inicialmente propostos e que esteja disponível no prazo mais curto possível. Este sendo o objetivo de cada empresa.

### **Documentação do desenvolvimento do projeto**

Este tópico apresenta a arquitetura de infra-estrutura do sistema, demonstrando o tipo de arquitetura orientada serviços bem como o dimensionamento mínimo de conexões.

A arquitetura da televisão baseada no *browser* é mostrada na Figura abaixo fazendo parte da documentação mandada pelo cliente. Como é representado na figura, a TV é constituída por duas partes principais; um portal e um dispositivo em que o portal é executado. O Portal é executado num *browser* que faz uso dos recursos disponíveis ou "Serviços de Sistema" do dispositivo. Quando as capacidades suficientes não estão disponíveis no dispositivo, os serviços para corrigir esta lacuna está instalado no dispositivo, representado pelo termo "Enabler Services".

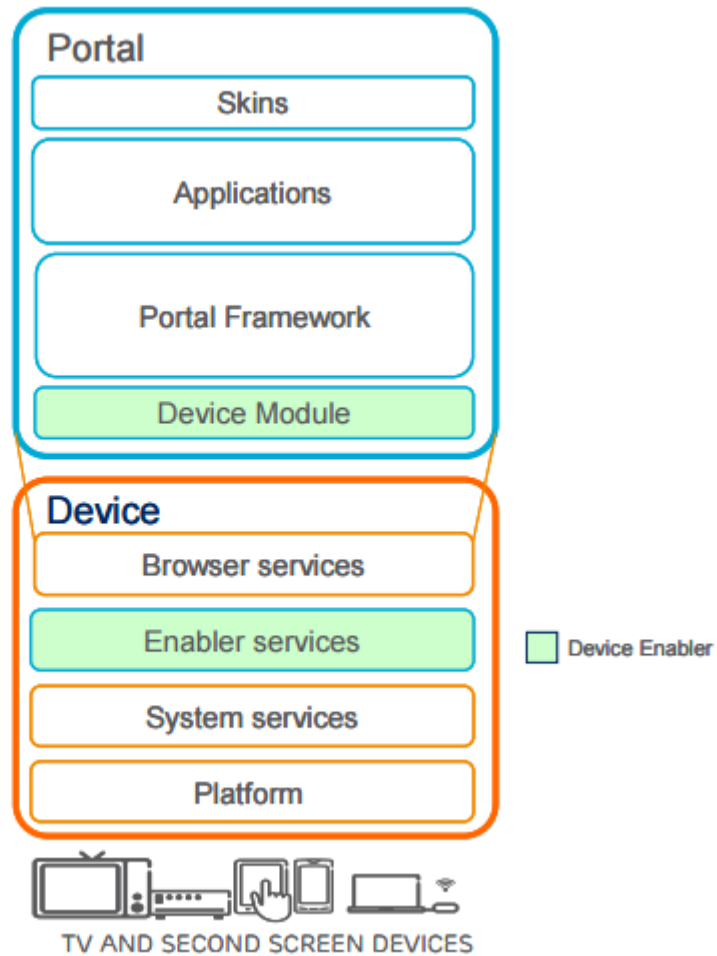


Figura 5. Architecture - Browser Based TV

O VOD ingestão num cenário *Multiscreen* (fazendo parte da documentação mandada pelo cliente) envolve a mesma conjunção de componentes com base em conteúdo de fluxo contínuo, com a exceção do *Encoder* para a transcodificação e segmentação. É possível ingerir um pacote de que inclui arquivos media para streaming. O *CMS* coordena a ingestão de fluxo em conjunto com o sistema de criptografia. Depois que o conteúdo é interferido, o conteúdo está disponível para compra por usuários.

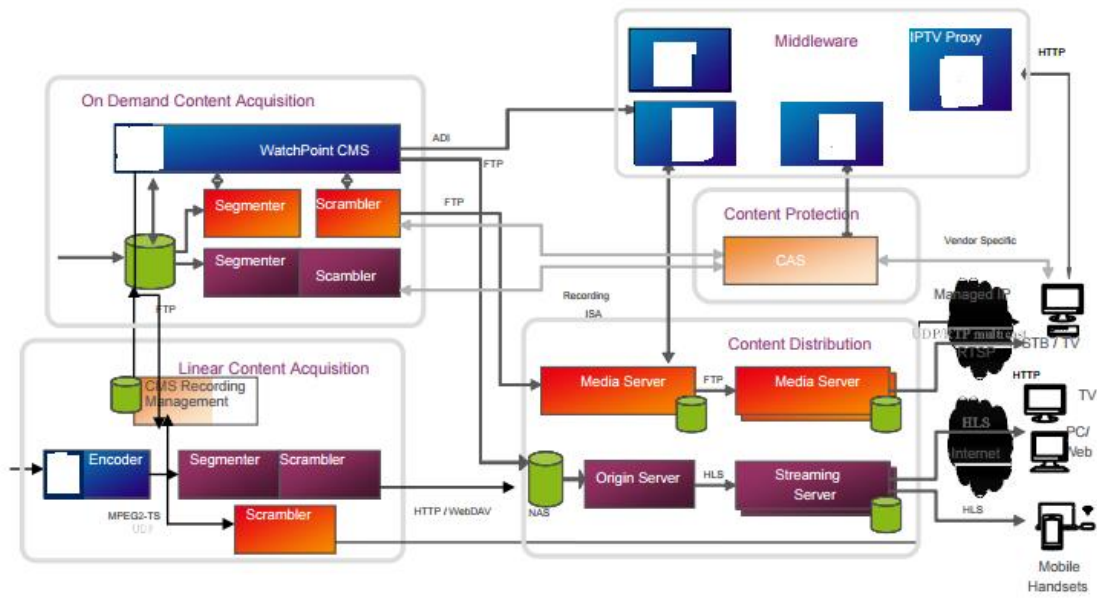


Figura 6. VOD Content with OpenStream

Antes de fazer as chamadas a *API*, deve-se aceder as informações do usuário. Este pode ser acedido no [api.softwareX.com/info](http://api.softwareX.com/info). Neste exemplo, a chamada é feita a partir de um país onde o SoftwareX ainda não está disponível, de modo a "abrir" o parâmetro que é definido como falso. Neste caso, o usuário deve estar pronta para entrar na aplicação. Se o parâmetro aberta é de verdade, então fazer o login é opcional.

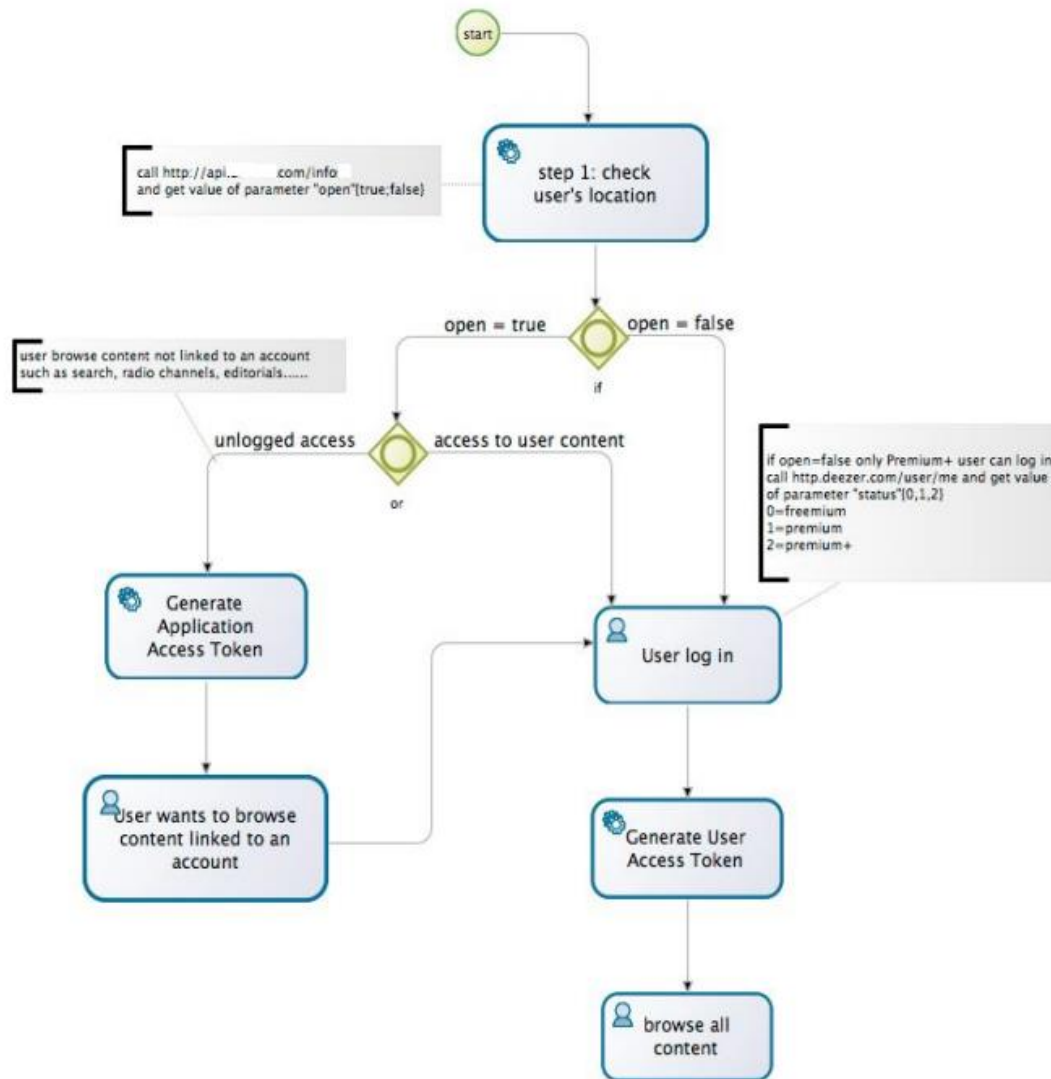


Figura 7. Content Workflow

## Documentação do Projeto

O processo de desenvolvimento do projeto durou 4 meses a partir de Abril de 2016 até Julho deste ano. O primeiro passo foi a reunião inicial do projeto, a entrega da documentação, inclusivamente os requisitos; o *design*; a programação; testes internos e a submissão as *stores Smart TV*. De seguida a versão do plano de desenvolvimento (Mapa de Gantt) do projeto SoftwareX efectuado pela GlobalSoftware.

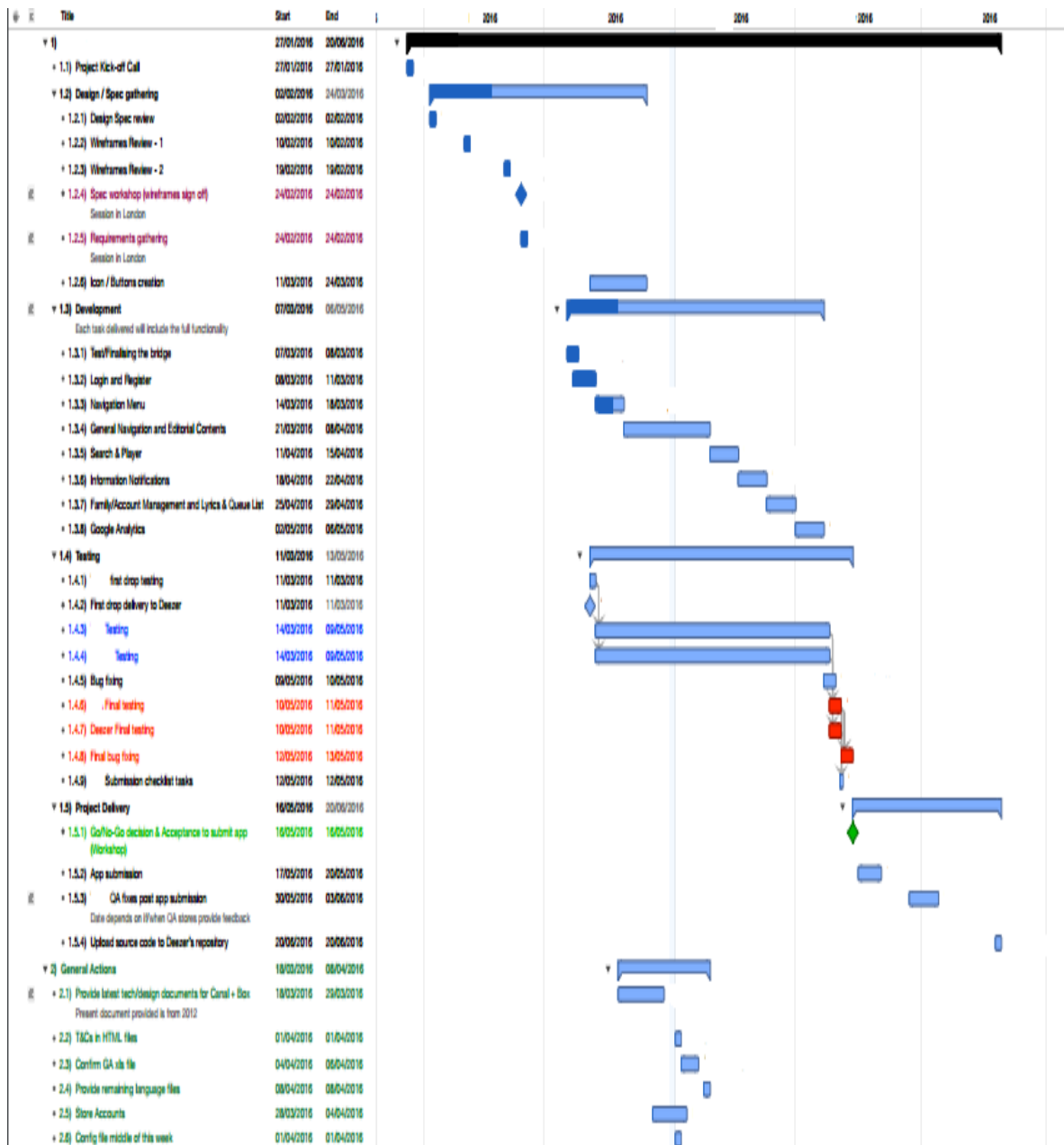


Figura 8. Mapa de Gantt SoftwareX

A figura 9 apresenta o *User Flow* (fazendo parte da documentação mandada pelo cliente), o diagrama que ilustra e permite a exploração a interface do user. Esse diagrama é usada para modelar as interações que os usuários têm com o seu *software*. Esta visão geral é efetivamente o resultado do ponto de vista arquitetônico da interface de usuário para compreender o sistema. Analisando a figura 9, ganhamos uma compreensão de como o sistema deverá trabalhar.

A aplicação tem várias camadas/páginas começando com o *splash screen*. A primeira página contém o logotipo, o login/registrar, a função trocar de conta, escolher o profile e entrar na página principal que contém o *Home*, a diversidade de música e a procura dentro da aplicação. A música está dividida em várias categorias e não só possibilita ouvi-la como também disponibiliza informação acerca do respectivo artista.

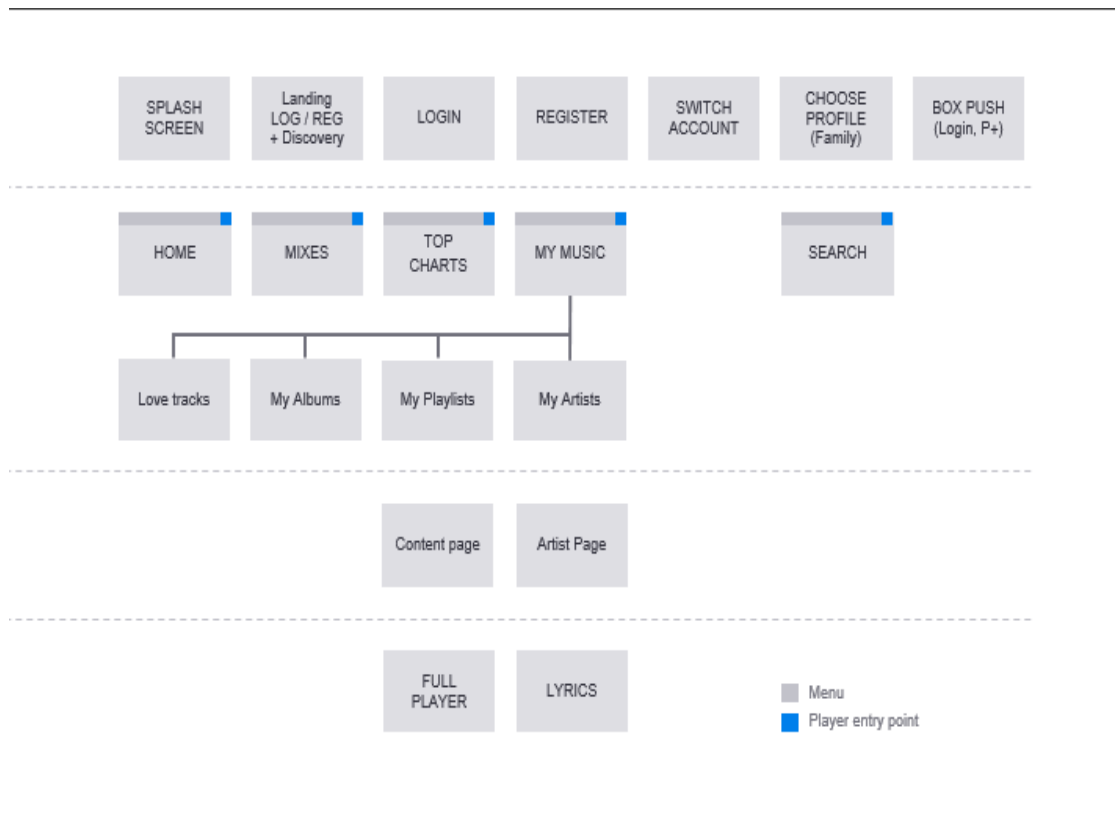


Figura 9. Diagrama *User Flow*

O diagrama de caso de uso descreve a funcionalidade proposta para o *login* do sistema que será projetado, é uma ferramenta para o levantamento dos requisitos funcionais do sistema. A Figura 10 apresenta a descrição da funcionalidade que será construída no login da aplicação (fazendo parte da documentação mandada pelo cliente).

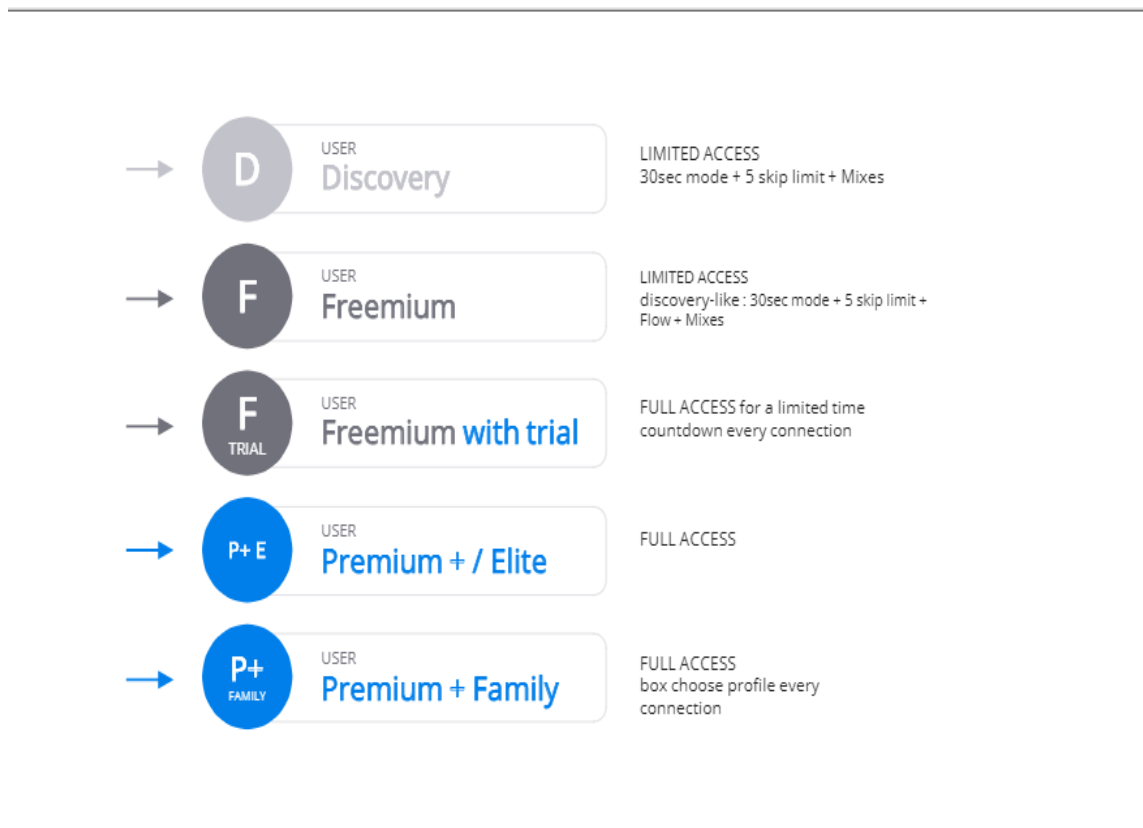


Figura 10. Diagrama casos de uso (Login)

Após o desenvolvimento do SoftwareX, segue a investigação do *software* a fim de fornecer informações sobre a qualidade em relação ao contexto em que ele deve operar. Isso inclui a validação da lista apresentada em baixo, uma guia para encontrar os defeitos. O teste é um processo realizado pelo testador e envolve a verificação dos requisitos inicialmente predefinidas. Este teste foi executado pela equipa de QA da GlobalSoftware na última etapa do projeto.

#### Planos de Testes QA

- Todos os *geoblocks* para testes estão desativados.
- A aplicação funciona normalmente, sem problemas ou erros.
- A aplicação funciona normalmente quando é lançada novamente depois de a fechar.
- Ao abrir a aplicação, o *loading cue* é visível até que o ecrã é carregado e é exibido normalmente.
- Se ocorrer um erro, uma mensagem é exibida ou de um *pop-up* de erro.
- Se pressionar o botão Sair no comando, deve exibir o *Live TV*.

- Se pressionar o botão Sair no ecrã, deve exibir o *Live TV*.
- O processo de registo pode ser concluído sem problemas ou erros.
- A operação de *Login* funciona corretamente.
- Os resultados da pesquisa são exibidos corretamente após a inserção de palavras de busca.
- Quando pressionado o botão OK do comando, funciona corretamente.
- Quando pressionada a tecla HOME do comando, funciona corretamente.
- Quando pressionado o botão *BACK* volta para a página anterior.
- Quando pressionado o botão EXIT do comando, a aplicação fecha normalmente e muda de volta para o *Live TV*.
- Os volumes do audio funcionam corretamente e estão em níveis adequados.
- Quando pressionado o botão *MUTE* do Comando geral o audio da aplicação fica em silêncio.
- O tamanho do texto e outros elementos devem ser ajustados em conformidade. (A legibilidade deve ser bem considerada).
- Deve ficar claro quem é o usuário conectado. Quando necessário, deve ser fácil para os usuários conectados entrarem novamente.
- As teclas de navegação (*Left, Right, Up, Down, Return, Enter*) funcionam sempre.
- Quando a internet é re-conectada, a aplicação funciona bem.

O último processo foi a criação de documentação para o usuário final para entender o funcionamento do sistema.

Em geral, o projeto foi bem-sucedido, mas concluído com um pouco de atraso para o orçamento previsto, apresentou eficiência na utilização dos recursos disponibilizados e atingiu a qualidade e performance desejada. Foi estabelecida uma metodologia scrum para a gestão, porém para mitigar qualquer tipo de problemas e para auxiliar as ferramentas e os processos. Foi testado várias vezes, os requisitos foram refinados e reavaliados e representaram mais fielmente as necessidades dos usuários. Com isso as oportunidades do projeto de desenvolvimento de *software* teve muito êxito.

# Capítulo 4

---

## GDD: Mitos e Realidades

Desde os primórdios da computação, a programação era visto como uma forma de arte, sem utilização de metodologias formais e sem qualquer preocupação com a documentação, entre outros fatores importantes. A experiência do programador era adquirida através de tentativa de erro.

A verdade é que esta tendência ainda se verifica. Com o crescimento dos custos de *software* no custo total de um sistema computacional e a incrível "resistência" às mudanças (particularmente, no que diz respeito ao uso de novas técnicas, o processo de desenvolvimento tornou-se um item de fundamental importância na produção de tais sistemas.

Dito isso, é importante ressaltar e discutir os chamados "mitos e realidades" do *software*, o que, de certo modo, explicam alguns dos problemas de desenvolvimento e entrega de software globalmente apresentados.

Os mitos referidos nos subcapítulos surgiram com base da minha investigação durante o estágio.

## 4.1 Mitos principais

As empresas reconhecem a necessidade de uma maior eficiência, menores custos de processamento e melhor visibilidade e controle dos processos de negócios sensíveis ao tempo a fim de capitalizar maiores economias de escala e conhecimento especializado. Tomando em conta as vantagens, as empresas ganham se processarem eficientemente o desenvolvimento global de *software*. No entanto, as empresas têm sido muitas vezes relutantes em alguns processos financeiros, tais como contas a pagar.

Mas será que a realidade é conhecida num ambiente global?

Neste tópico levanto algumas particularidades, algumas questões que continuam mitos e verdades falsas no ponto de vista da maioria.

**Mito:** #1

Mito: Não existe solução exata para cada problema específico em GDD.

Credibilidade: Intermédia

Realidade: As soluções são relevantes para todos os modos de colaboração. As soluções são apresentadas a partir dos três pontos de vista da tecnologia, processos e pessoas e para cada uma das causas apresentadas de seguida.

As soluções tecnológicas não são de todo relevantes para os desafios causados pelas diferentes origens e conhecimento tácito ou que estas soluções não são relevantes para os desafios causados pela diferença de tempo, distância também como a falta de comunicação. O ponto está em mostrar que tipos de soluções são mais relevantes para os desafios com origem em por várias causas. As soluções de tecnologia geralmente podem suportar os desafios associados à diferença de tempo e distância e falta de comunicação, mas também, em conjunto com soluções de processo, eles também podem apoiar para além de outros, desafios causados por várias partes / partes interessadas e discriminação na coordenação. As soluções relacionadas

com as pessoas podem especialmente apoiar desafios causados por diferentes formações e conhecimento tácito. Relativamente à falta de comunicação as soluções também são relevantes, mas normalmente precisam também de tecnologia ou processo de apoio no GDD (i.e., ferramentas de comunicação, ou a partilha de informação prática), as soluções de processo os poderem suportar. Em seguida, as soluções são discutidas de acordo com os três pontos de vista. Em primeiro lugar, as soluções são discutidos em geral e, em seguida, exemplos de soluções são apresentados em tabelas utilizando a estrutura no quadro de GDD.

#### 1. Solução de processo

As soluções relacionadas com o processo podem ajudar a abordar todas as causas para os problemas e desafios na GDD. Uma solução é geralmente uma solução parcial para o problema e deve ser utilizado em combinação com outras soluções para resolver completamente o problema. Não existe um modelo específico para os processos disponíveis, que iriam abordar GDD como um todo. Tal modelo também não seria prático, já que normalmente requer uma mudança completa na maneira de trabalhar dentro de uma empresa. Em vez disso, novas práticas ou melhorias de práticas são propostas. Estas práticas podem ser incorporados em qualquer modelo de processo, mas os detalhes de como resolvê-los na prática podem variar. Por exemplo, o Välimäki [44] propõe uma abordagem baseada em padrões para integrar as boas práticas que suportam GDD no processo da empresa e [45] discutem como as práticas ágeis podem ajudar na GDD. As soluções são discutidas aqui de acordo com a estrutura dos tópicos: **Práticas de gestão, Práticas de engenharia e Práticas de suporte.**

- **Práticas de gestão**

As práticas de gestão incluem atividades a nível organizacional e atividades a nível de projeto. Estas atividades são as ferramentas necessárias para o sucesso no GDD. Além disso são a base como apresentado, por exemplo, em estruturas de gestão de projetos, como no PMBOK.

As práticas de gestão incluem a gestão de alteração de requisitos que permitem a coerência entre os requisitos e outras partes do *software* servindo de um meio de comunicação entre a equipa e os projetos.

As práticas de gestão resumem os desafios e soluções de processo relacionadas com cada um dos seguintes tópicos apresentados abaixo.

### Estratégia de colaboração:

Em primeiro lugar, uma estratégia de colaboração deve incluir modelos aplicáveis á colaboração. A empresa deve ter uma estratégia para cada situação, considerando também a estrutura organizacional e a cultura. Também deve definir uma política de recompensa para os esforços da colaboração a fim de incentivar, por exemplo, apoiando e ajudando uns aos outros. A colaboração também deve ser levado em conta no planeamento a longo prazo.

Problemática	Solução
<ul style="list-style-type: none"><li>• <b>Colaboração ineficaz devido ao modo de colaboração inadequada.</b></li><li>• <b>Atrasos e problemas em cooperação devido à forma de trabalhar.</b></li></ul>	<ul style="list-style-type: none"><li>• Práticas para a seleção de modo de colaboração.</li><li>• Características do modo de colaboração.</li><li>• Definição de organização do projeto proposital e as práticas de colaboração apropriadas para o modo de colaboração.</li><li>• Envolvimento dos parceiros no mapeamento para estabelecer a confiança e trabalho cooperativo.</li></ul>

### Práticas de contratação

As práticas de contratação é outro tema importante a nível da gestão e práticas, especialmente nos esforços de colaboração global. As questões como a garantia que os contratos sejam assinados no início do empreendimento cooperativo envolvendo as pessoas técnicas no processo de contratação para garantir que os acordos tecnicamente realistas são feitos, assi como concordar com os procedimentos de alteração que os pontos de decisão devem ser incluídos no processo de contratação. Durante o processo de contratação todas as partes envolvidas ou partes interessadas, com as autoridades necessárias, devem ser envolvidos. As condições para a colaboração devem ser estabelecidas para a organização, bem como para cada projeto distribuído. Estas condições incluem a comunicação, motivação e razão para a colaboração de todas as partes, estabelecendo uma cultura de colaboração que se torna uma

forma natural de trabalho, identificando as diferenças culturais entre os parceiros e dividindo o trabalho em locais distribuídos.

Os tópicos discutidos nesta seção são relevantes no que diz respeito à criação de condições para a colaboração. Além disso, definindo a confidencialidade dos dados e permitindo ou

Problemática	Solução
<ul style="list-style-type: none"><li>• <b>Acordos obscuros ou inexistentes entre os parceiros.</b></li><li>• <b>Lacunas e / ou itens duplicados na cobertura dos contratos.</b></li></ul>	<ul style="list-style-type: none"><li>• Práticas para o estabelecer bons contratos ou modelos de contrato.</li><li>• Práticas para aceitar questões não incluídas na gestão de mudanças no contrato.</li></ul>

restringindo o acesso, a fim de assegurar que a informação requerida esteja disponível para todos aqueles que precisam dela.

#### Condições para colaboração:

Neste tópico apresento as práticas que têm de ser trabalhadas no início do projeto para assegurar boas condições para assim o começar, por exemplo, a identificação das diferenças culturais.

Problemática	Solução
<ul style="list-style-type: none"><li>• <b>Atrasos e esforço desperdiçado devido à inadequada organização do projeto ou práticas e ferramentas de colaboração em falta.</b></li><li>• <b>Falta de visibilidade do trabalho do parceiro.</b></li></ul>	<ul style="list-style-type: none"><li>• Práticas para a partilha de informações e a gestão da mudança.</li><li>• As responsabilidades e autoridades definidas e pessoas de contacto para cada local.</li><li>• Comunicação frequente.</li></ul>

#### A gestão de fornecedores:

Neste tópico descrevo as práticas que os fornecedores devem estar relacionados em vários pontos do projeto e a atenção para a deteção de potenciais problemas no início do mesmo.

Problemática	Solução
<p><b>Duplicação de trabalho causado por atribuições pouco claras, os atrasos causados por dependências não gerenciados.</b></p> <p><b>Não ou não bastante visibilidade do trabalho dos fornecedores.</b></p> <p><b>Entregas inesperados (conteúdo ou tempo).</b></p> <p><b>Requisitos mal compreendidos.</b></p>	<p>Práticas de gestão de fornecedores, incluindo responsabilidades, <i>checkpoints</i>, contatos e tarefas de alinhamento.</p> <p>Práticas fornecedor-seleção definidos tendo também em conta a experiência de colaboração dos fornecedores.</p> <p>Atenção explícita colocada sobre a gestão dos fornecedores.</p>

Gestão de projetos:

A gestão de projetos requer uma atenção específica para os projetos globalmente distribuídos. As funcionalidades dos mesmos devem ser definidas propositadamente no que respeita às situações, o que significa que o trabalho é distribuído e as dependências entre os locais são minimizados, ainda utilizando as melhores competências virtuais para várias tarefas. Os objetivos dos projetos precisam de ser explícitos, sem ambiguidades e comunicados a todas as equipas envolvidas para criar confiança mútua.

Um aspeto importante relativo à gestão do projeto GDD é o alinhamento entre as equipas e

Problemática	Solução
<ul style="list-style-type: none"> <li>• Estado não bem especificado de projeto.</li> <li>• Atrasos causados por autoridades e práticas tomadas da decisão pouco clara.</li> <li>• Atrasos causados por equipas não-alinhadas.</li> <li>• Resultados do trabalho não correspondente causados por objetivos incompreendidos.</li> <li>• Recursos indisponíveis.</li> </ul>	<ul style="list-style-type: none"> <li>• Definição de um plano de projeto e estrutura de gestão, incluindo as tarefas, responsabilidades, dependências e esforço planeado e horários.</li> <li>• Práticas para o alinhamento das equipas e parceiros de trabalho, práticas para a gestão de dependências entre os parceiros, gestão de recursos críticos.</li> </ul>

parceiros, o que significa que as dependências entre mesmas são identificados e administrados de uma forma proactiva. A comunicação é um aspeto importante, os itens de comunicação, os papéis e as ferramentas apropriadas devem ser definidas inicialmente.

- **Práticas de engenharia**

Há uma série de soluções disponíveis especialmente para as práticas de engenharia. Em seguida algumas soluções apresentadas.

#### Engenharia de requisitos

Este tópico descreve as práticas para a identificação dos requisitos ambíguos ou as diferenças na compreensão das exigências entre parceiros.

<b>Problemática</b>	<b>Solução</b>
<ul style="list-style-type: none"><li>• <b>Documentação não detalhada.</b></li><li>• <b>Dificuldades na criação e gestão de rastreabilidade.</b></li><li>• <b>Requisitos mal compreendidos causando entregas inconvenientes.</b></li><li>• <b>Problemas causados por mudanças não comunicadas com os requisitos.</b></li><li>• <b>As inconsistências entre requisitos e outros produtos de trabalho.</b></li></ul>	<ul style="list-style-type: none"><li>• Requisitos práticos de priorização.</li><li>• Modelo de documento que abrange aspetos de qualidade de bons requisitos.</li><li>• Requisitos do processo de engenharia para a configuração distribuída.</li><li>• Práticas para identificar e lidar com os requisitos ambíguos para a resolução de requisitos conflitantes.</li><li>• Práticas de gestão de requisitos uniformes entre as equipas, incluindo requisitos claros.</li></ul>

#### Arquitetura

Nesse tópico são apresentadas as práticas para assegurar um entendimento comum da arquitetura entre os parceiros, por exemplo, através da utilização de várias visões de arquitetura e princípios de *design*.

Problemática	Solução
<ul style="list-style-type: none"><li data-bbox="193 259 703 344">• <b>Arquitetura inadequada (não apoia o desenvolvimento distribuído).</b></li><li data-bbox="193 367 687 452">• <b>Não está claro, não-comunicativa ou arquitetura incompreendida.</b></li></ul>	<ul style="list-style-type: none"><li data-bbox="799 259 1286 452">• Práticas de definição de arquitetura, incluindo lista de verificação para a definição de arquitetura para desenvolvimento distribuído.</li><li data-bbox="799 474 1318 560">• Manutenção e evolução da arquitetura claramente definida.</li><li data-bbox="799 582 1358 667">• Tendo em conta os modos de colaboração na definição da arquitetura.</li></ul>

### Design e Código

Neste tópico, apresento a descrição das regras comuns de *design* e implementação que devem ser seguidas pelos parceiros, a fim de garantir uma qualidade comum e compreensibilidade do trabalho feito por outros; por exemplo os padrões de desenho a serem utilizados.

Problemática	Solução
<ul style="list-style-type: none"><li>• <b>Dificuldades de comunicação causando erros em decisões de <i>design</i>.</b></li><li>• <b>Estilos de codificação diferentes e normas que afetam a compreensibilidade do <i>design</i> e código por outros.</b></li></ul>	<ul style="list-style-type: none"><li>• Práticas de revisão para garantir que as escolhas de <i>design</i> são boas.</li><li>• Verificação para garantir a qualidade do padrão de desenho e codificação.</li></ul>

### Integração

Neste tópico apresento a descrição das diferentes estratégias de integração, que ajudam a escolher o caminho mais certo para o projeto.

Problemática	Solução
<ul style="list-style-type: none"><li>• <b>Os atrasos nas respostas para os problemas encontrados na integração, indisponibilidade de informações.</b></li><li>• <b>Problemas na integração causados pelas entregas não sincronizadas.</b></li></ul>	<ul style="list-style-type: none"><li>• Responsabilidades para a integração e resolução de problemas definidas.</li><li>• Uma boa estratégia de integração.</li><li>• Práticas de integração contínua.</li></ul>

### Testes

Neste tópico apresento as práticas para apoiar a partilha de informações de teste (casos de teste, os resultados do teste, dados de teste) entre os membros das equipas.

Problemática	Solução
<ul style="list-style-type: none"> <li>• Dificuldades na replicação dos defeitos encontrados nos testes.</li> <li>• Falta de testes completos.</li> <li>• Testes inadequados (devido à falta de conhecimento).</li> </ul>	<ul style="list-style-type: none"> <li>• Definidas práticas de teste para configurações distribuídas, responsabilidades definidas.</li> <li>• Partilha de casos de teste.</li> <li>• Práticas para compartilhar informações sobre os testes realizados e os resultados do teste.</li> </ul>

- **Práticas de suporte**

O Herbsleb e Moitra [14] apontam que a gestão de configuração envolve a transmissão de dados críticos e de produção em vários locais que deve ser bem planeados e executados. Uma possível solução é uma ferramenta de gestão de configuração de *software* comum com a replicação em vários *sites* e um repositório centralizado a ser útil no desenvolvimento distribuído.

### Gestão de configuração

Este tópico descreve as práticas de gestão de configuração que devem ser comuns entre todos os parceiros.

Problemática	Solução
<ul style="list-style-type: none"> <li>• Práticas de gestão de configuração complexa.</li> <li>• Dificuldades em encontrar versões corretas.</li> <li>• Uso de versões erradas causando problemas em compilações.</li> <li>• Estado do trabalho de desenvolvimento não especificado.</li> </ul>	<ul style="list-style-type: none"> <li>• Processo de gestão de configuração de suporte de desenvolvimento distribuído, incluindo a comunicação de alterações de papéis e responsabilidades.</li> </ul>

### A garantia de qualidade

Este tópico descreve as práticas de gestão de qualidade que devem ser comuns entre todos os parceiros.

<b>Problemática</b>	<b>Solução</b>
<ul style="list-style-type: none"><li>• <b>Práticas diferentes e desconhecidas de parceiros.</b></li><li>• <b>Responsabilidades desconhecidas para a garantia da qualidade.</b></li><li>• <b>As diferenças de qualidade do trabalho entre os parceiros.</b></li></ul>	<ul style="list-style-type: none"><li>• Processo compartilhado entre os parceiros para certas partes, como gestão de configuração, gestão de mudanças, gestão de requisitos e gestão de projetos.</li><li>• Interfaces definidas para os parceiros.</li><li>• Processo comum para os itens relevantes em sites / parceiros.</li><li>• Gestão de riscos relacionados com a colaboração.</li></ul>

### A partilha de informação

Este tópico descreve as práticas de documentação que tornam o compartilhamento de informações em projetos distribuídos mais fácil (compreensibilidade e disponibilidade de documenta

<b>Problemática</b>	<b>Solução</b>
<ul style="list-style-type: none"><li>• <b>Dificuldades em saber quem tem a informação relevante e quando necessário.</b></li><li>• <b>As dificuldades em definir os meios adequados para a partilha de informação.</b></li><li>• <b>Dificuldades em saber que informação deve ser comunicada e como (o que já é conhecido e o que não é).</b></li><li>• <b>Responsabilidades pouco claras para o compartilhamento de informações.</b></li></ul>	<ul style="list-style-type: none"><li>• Práticas definidas de partilha de informações, incluindo o quê, quando, de quem, para quem e com que meios.</li><li>• Contactos definidos de cada parceiro.</li><li>• Acordo entre parceiros sobre a forma como as mudanças afetam os contratos.</li><li>• Práticas para avaliar o impacto da mudança sobre o trabalho de outros partidos.</li></ul>

### A partilha de informação

Este tópico descreve as práticas de documentação que tornam o compartilhamento de informações em projetos distribuídos mais fácil (compreensibilidade e disponibilidade de documentação).

<b>Problemática</b>	<b>Solução</b>
<b>Dificuldades em saber quem tem a informação relevante e quando necessária.</b>	Práticas definidas de partilha de informações, incluindo o quê, quando, de quem, para quem e com que meios.
<b>As dificuldades em definir os meios adequados para a partilha de informação.</b>	Contatos definido de cada parceiro. Acordo entre parceiros sobre a forma como as mudanças afetam os contratos.
<b>Dificuldades em saber que informação deve ser comunicada e como (o que já é conhecido e o que não é).</b>	Práticas para avaliar o impacto da mudança sobre o trabalho de outros partidos.
<b>Responsabilidades pouco claras para o compartilhamento de informações.</b>	

### Infraestrutura

Este tópico apresenta a lista de verificação que ajuda a garantir que a infra-estrutura é adequada para qualquer situação em questão.

<b>Problemática</b>	<b>Solução</b>
<b>Diferentes ferramentas incompatíveis utilizados pelos parceiros.</b>	Definir uma só forma de trabalhar com a infraestrutura.
<b>Problemas na disponibilidade de ferramentas.</b>	Maneira definida de trabalhar com ferramentas
<b>Diferentes maneiras de usar ferramentas.</b>	Práticas de reuniões virtuais.
<b>Incapacidade de usar ferramentas de comunicação de forma eficaz.</b>	

### Gestão de competência

Este tópico apresenta as práticas de desenvolvimento da equipa a considerar quando construímos uma equipa, por exemplo, as competências adequadas e construir uma atmosfera amigável.

<b>Problemática</b>	<b>Solução</b>
<ul style="list-style-type: none"><li>• <b>O desconhecimento das competências.</b></li><li>• <b>Utilização não otimizada dos recursos.</b></li></ul>	<ul style="list-style-type: none"><li>• Práticas de analisar as competências necessárias</li><li>• Práticas para a utilização das competências da melhor forma.</li></ul>

### Melhoria contínua

Esse tópico apresenta as métricas e práticas de medição que fornecem informações sobre a eficácia da colaboração durante e depois do projecto.

<b>Problemática</b>	<b>Solução</b>
<ul style="list-style-type: none"><li>• <b>Práticas diferentes e desconhecidas de parceiros, as responsabilidades pouco claras e autoridades para o trabalho de melhoria.</b></li><li>• <b>Informações não partilhadas sobre as lições aprendidas e as melhores práticas.</b></li></ul>	<ul style="list-style-type: none"><li>• Práticas para compartilhar as melhores práticas e lições aprendidas.</li><li>• Avaliar a eficácia da colaboração.</li></ul>

## 2. Soluções de tecnologia

As soluções tecnológicas para GDD são principalmente úteis para enfrentar os desafios causados pela diferença de tempo, distância e falta de comunicação. Além disso, as soluções de tecnologia precisam de um processamento ou pessoas relacionadas com soluções para apoiá-los. Por exemplo, a disponibilidade de ferramentas de comunicação não significa automaticamente que a comunicação ocorre, mas a sua ausência dificulta a comunicação. As soluções de tecnologia são principalmente ferramentas e questões de infraestrutura; por exemplo, redes e ferramentas que podem ajudar na partilha de informação entre os parceiros.

As ferramentas suficientes para a comunicação e repositórios compartilhados também são muito importantes para o sucesso GDD. Os recursos devem ser acessíveis a todos os que necessitam e devem ser compatíveis entre os parceiros, quando necessário. Além disso, a conexão de rede deve ser utilizável e confiável. A atenção especial deve ser dada aos problemas causados pela diferença de horário; por exemplo, as ferramentas apropriadas para o compartilhamento de informações estarem disponíveis e que a transição entre o trabalho síncrona e assíncrona é gerido. As ferramentas utilizadas na GDD são as mesmas utilizadas num único local com a adição de ferramentas de comunicação e repositórios partilhados. Os aspetos importantes ao escolher as ferramentas são os requisitos normais para ferramentas, tais como a usabilidade e aptidão para o efeito, mas elas têm que ser consideradas do ponto de vista GDD, o que significa que múltiplos parceiros com diferentes formações precisam ser levados em conta. A avaliação sistemática e a seleção de ferramentas para projeto o GDD é importante, por exemplo, Poston e Sexton [46] apresentam um fluxo de trabalho para a avaliação da ferramenta em relação a ferramentas de teste que foi prorrogado por [18] para as avaliar para os projetos distribuídos. Esse fluxo de trabalho consiste em cinco etapas: análise e classificação de requisitos (1), busca e categorização de ferramentas (2), métodos de avaliação (3), cenário de *brainstorming* (4), análise e avaliação de ferramentas (5), e pode ser utilizado na avaliação de outras ferramentas para o GDD.

- **Práticas de gestão**

As práticas de gestão resumem os desafios e soluções de tecnologia relacionada para cada um dos seguintes tópicos apresentados a seguir.

Estratégia de colaboração

Este tópico descreve as principais características dos modelos de colaboração selecionados, a fim de ao escolhê-las ajudar a compreender os diferentes modos.

Problemática	Solução
<ul style="list-style-type: none"> <li>• <b>Colaboração ineficaz devido ao modo de colaboração inadequada.</b></li> <li>• <b>Atrasos e problemas em cooperação devido à forma de trabalhar.</b></li> </ul>	<ul style="list-style-type: none"> <li>• Fóruns de discussão para perguntas e respostas (que são armazenados); Adicionar os membros do projeto (com fotos).</li> </ul>

### Condições para colaboração

Este tópico apresenta as práticas sugeridas a serem introduzidas desde do início do projeto para assegurar as boas condições, por exemplo, a identificação de diferenças culturais.

<b>Problemática</b>	<b>Solução</b>
<ul style="list-style-type: none"><li>• <b>Atrasos e esforço desperdiçado devido à inadequada organização do projeto ou práticas e ferramentas de colaboração em falta.</b></li><li>• <b>Falta de visibilidade do trabalho dos parceiros.</b></li></ul>	<ul style="list-style-type: none"><li>• O estabelecimento de infraestrutura de trabalho, incluindo gestão de configuração, partilhada, compartilhamento de documentos e ferramentas de comunicação entre os membros.</li><li>• Acesso a ferramentas para obter informações em tempo real.</li></ul>

### Gestão de fornecedores

Este tópico descreve que os fornecedores devem estar relacionados em vários pontos do projeto e a atenção para a deteção de potenciais problemas desde o início.

<b>Problemática</b>	<b>Solução</b>
<ul style="list-style-type: none"><li>• <b>Duplicação de trabalho causado por atribuições pouco claras, os atrasos causados por dependências não gerenciados.</b></li><li>• <b>Não ou não bastante visibilidade do trabalho dos fornecedores.</b></li><li>• <b>Entregas inesperados (conteúdo ou tempo).</b></li><li>• <b>Requisitos mal compreendidos.</b></li></ul>	<ul style="list-style-type: none"><li>• Acesso a ferramentas de fornecedores de informações em tempo real.</li></ul>

### Gestão de projetos

Este tópico apresenta a descrição das metas de ponto de vista GDD para acompanhar o andamento do projeto e as orientações para reagir aos desvios ao atingir essas metas.

<b>Problemática</b>	<b>Solução</b>
<ul style="list-style-type: none"> <li>• Estado de projeto não especificado.</li> <li>• Atrasos causados por autoridades e práticas de tomada de decisões pouco claras.</li> <li>• Atrasos causados por equipas não-alinhadas.</li> <li>• Objetivos incompreendidos.</li> <li>• Recursos indisponíveis.</li> </ul>	<ul style="list-style-type: none"> <li>• Ferramentas de gestão de projeto ao longo de <i>sites</i>, como o esforço, o progresso, e informações do projeto.</li> </ul>

- **Práticas de engenharia**

Há uma série de soluções disponíveis especialmente para as práticas de engenharia. Em seguida algumas soluções apresentadas.

Engenharia de requisitos

Este tópico descreve as práticas para a identificação dos requisitos ambíguos ou as diferenças na compreensão das exigências entre parceiros.

<b>Problemática</b>	<b>Solução</b>
<ul style="list-style-type: none"> <li>• Documentação pouco clara.</li> <li>• Dificuldades na criação e gestão dos requisitos.</li> <li>• Requisitos mal compreendidos.</li> <li>• Problemas causados por mudanças não comunicadas com os requisitos.</li> <li>• As inconsistências entre requisitos e outros produtos de trabalho.</li> </ul>	<ul style="list-style-type: none"> <li>• Ferramentas de engenharia que suportam trabalho multi-site.</li> <li>• Requisitos compartilhados para assegurar a disponibilidade de versões corretas e mudanças.</li> </ul>

Arquitetura

Nesse tópico as práticas para assegurar um entendimento comum da arquitetura entre os parceiros, por exemplo, através da utilização de várias visões de arquitetura e princípios de *design*.

Problemática	Solução
<ul style="list-style-type: none"> <li>• Arquitetura inadequada (não apoiar o desenvolvimento distribuído)</li> <li>• Arquitetura não clara ou não comunicada.</li> </ul>	<ul style="list-style-type: none"> <li>• Fórum de discussão para compartilhar a informação arquitetural.</li> </ul>

### O design e o código

Neste tópico, apresento a descrição das regras comuns de *design* e implementação que devem ser seguidas pelos parceiros, a fim de garantir uma qualidade comum e compreensibilidade do trabalho feito por outros; por exemplo os padrões de desenho a serem utilizados.

Problemática	Solução
<ul style="list-style-type: none"> <li>• Dificuldades de comunicação causando erros em decisões de <i>design</i>.</li> <li>• Estilos de programação diferentes e normas que afetam a compreensibilidade do <i>design</i> e código dos outros.</li> </ul>	<ul style="list-style-type: none"> <li>• Repositório compartilhado com as versões de projeto corretas e decisões de <i>design</i> racional disponíveis.</li> <li>• Fórum de discussão para compartilhar a conceção lógica.</li> </ul>

### Integração

Neste tópico apresento a descrição das diferentes estratégias de integração, que ajudam a escolher o caminho mais certo para o projeto.

Problemática	Solução
<ul style="list-style-type: none"> <li>• Os atrasos nas respostas para os problemas encontrados na integração e indisponibilidade de informações.</li> <li>• Problemas na integração causados pelas partes não sincronizadas.</li> </ul>	<ul style="list-style-type: none"> <li>• Ferramenta de gestão de configuração.</li> <li>• Fórum de discussão para compartilhar os problemas de integração.</li> <li>• Ferramenta de gestão de defeitos para se comunicar e controlar defeitos.</li> <li>• Ferramentas de integração e</li> </ul>

conjuntos de testes automatizados compartilhados que fornecem feedback imediato para os desenvolvedores.

## Testes

Neste tópico apresento as práticas para apoiar a partilha de informações de teste (casos de teste, os resultados do teste, dados de teste) entre os membros das equipas.

Problemática	Solução
<ul style="list-style-type: none"><li>• <b>Dificuldades na duplicação dos defeitos encontrados nos testes.</b></li><li>• <b>Falta de visibilidade em testes.</b></li><li>• <b>Testes inadequados (devido à falta de conhecimento).</b></li><li>• <b>Teste ineficaz causado por partes não sincronizadas.</b></li></ul>	<ul style="list-style-type: none"><li>• Gestão de configuração compartilhada</li><li>• Sistema para garantir o uso da versão correta.</li><li>• Ambiente de teste partilhado.</li></ul>

- **Práticas de suporte**

As práticas de suporte devem ser ajustadas para apoiar o fenómeno GDD.

As melhores práticas devem ser registadas e usadas entre os parceiros. Além disso, em relacionamentos a longo prazo, é útil o trabalho de processos de melhoria. De seguida encontra-se uma amostra de soluções para apoiar as práticas.

## Gestão de configuração

Este tópico descreve as práticas de gestão de configuração que devem ser comuns entre todos os parceiros.

Problemática	Solução
<ul style="list-style-type: none"> <li>• Práticas de gestão de configuração complexa.</li> <li>• Dificuldades em encontrar versões corretas.</li> <li>• Utilização de versões erradas, causando problemas em compilações.</li> <li>• Estado não especificado do trabalho de desenvolvimento.</li> </ul>	<ul style="list-style-type: none"> <li>• Ferramenta de gestão de configuração que suporta desenvolvimento distribuído.</li> <li>• <i>Sites</i> compartilhados para distribuir informações.</li> </ul>

### A garantia de qualidade

Este tópico descreve as práticas de gestão de qualidade que devem ser comuns entre todos os parceiros.

Problemática	Solução
<ul style="list-style-type: none"> <li>• Práticas diferentes e desconhecidas de parceiros.</li> <li>• Responsabilidades desconhecidas para a garantia da qualidade.</li> <li>• Diferenças da qualidade do trabalho entre os parceiros.</li> </ul>	<ul style="list-style-type: none"> <li>• Repositórios compartilhados para resultados de controlo de qualidade.</li> </ul>

### A partilha de informação

Este tópico descreve as práticas de documentação que tornam o compartilhamento de informações em projetos distribuídos mais fácil (compreensibilidade e disponibilidade de documentação).

Problemática	Solução
<ul style="list-style-type: none"> <li>• Dificuldades em saber quem tem a informação relevante.</li> <li>• As dificuldades em definir os meios adequados para a partilha de informação.</li> </ul>	<ul style="list-style-type: none"> <li>• Repositórios compartilhados e ferramentas de comunicação funcionais.</li> <li>• Ferramentas de gestão de mudança.</li> </ul>

- Dificuldades em saber que informação deve ser comunicada e como (o que já é conhecido e o que não é).
- Responsabilidades pouco claras para o compartilhamento de informações.

### Infraestrutura

Este tópico apresenta a lista de verificação que ajuda a garantir que a infra-estrutura é adequada para qualquer situação em questão GDD.

<b>Problemática</b>	<b>Solucao</b>
<ul style="list-style-type: none"> <li>• Diferentes ferramentas incompatíveis utilizados pelos parceiros.</li> <li>• Problemas na disponibilidade de ferramentas.</li> <li>• Diferentes maneiras de usar ferramentas.</li> <li>• Incapacidade de usar ferramentas de comunicação de forma eficaz.</li> </ul>	<ul style="list-style-type: none"> <li>• Soluções de interoperabilidade para as ferramentas.</li> <li>• Ferramenta compartilhada para itens de trabalho compartilhados.</li> </ul>

### Gestão de competências

Estes tópicos apresentam as práticas de desenvolvimento da equipa a considerar quando construímos, por exemplo, as competências adequadas para criar uma atmosfera amigável.

<b>Problemática</b>	<b>Solução</b>
<ul style="list-style-type: none"> <li>• O desconhecimento das competências.</li> <li>• Utilização não otimizada dos recursos.</li> <li>• Competências necessárias não disponíveis</li> </ul>	<ul style="list-style-type: none"> <li>• Repositório compartilhado das competências e suas informações relacionadas com o projeto.</li> </ul>

### Melhoria contínua

Esse tópico apresenta as métricas e práticas de medição que fornecem informações sobre a eficácia da colaboração durante e depois do projecto.

Problemática	Solução
<ul style="list-style-type: none"> <li>• <b>Práticas diferentes e desconhecidas de parceiros e as responsabilidades pouco claras para a melhoria do trabalho.</b></li> <li>• <b>Informação não partilhada sobre as lições aprendidas e as melhores práticas.</b></li> </ul>	<ul style="list-style-type: none"> <li>• Repositório compartilhado de melhores práticas e lições aprendidas.</li> </ul>

### 3. Soluções em recursos humanos

As soluções em pessoas são principalmente úteis para enfrentar os desafios causados pelas diferentes origens e conhecimento tácito. As soluções nesta categoria estão relacionadas essencialmente com as competências e formação do pessoal, a comunicação, motivação e atitude das pessoas para o trabalho cooperativo, incluindo a identificação das diferenças culturais a levá-las em conta.

Durante o projeto GDD, uma grande quantidade de conhecimento é criado e precisa de ser compartilhado entre os parceiros. Assim, são necessárias soluções que aumentem a comunicação, confiança e consciência, bem como soluções que tornem o conhecimento disponível de forma explícita. A partilha de conhecimento pode ser apoiada por atividades relacionadas com a negociação, brainstorming e livremente chegar a um consenso de que também pode ajudar na criação de uma melhor compreensão mútua. Em GDD, a compreensão individual e compartilhada, ou seja, a criação de conhecimento requer muita comunicação, pois esta aumenta a confiança mútua entre os parceiros. Além disso, o papel da comunicação informal é importante, pois ajuda a criar confiança e *teamness* entre os parceiros. O estabelecimento de uma cultura de colaboração pode ser suportada pelas reuniões cara-a-cara, práticas de construção de equipa, recompensando os esforços de colaboração, e atribuindo tarefas e responsabilidades explícitas para a comunicação através de *sites*.

- **Práticas de gestão**

As práticas de gestão resumem os desafios e soluções de recursos humanos associados a para cada um dos seguintes tópicos a seguir.

### Estratégia de colaboração

Este tópico descreve as principais características dos modelos de colaboração selecionados, a fim de ao escolê-los ajudar a compreender os diferentes modos.

<b>Problemática</b>	<b>Solução</b>
<ul style="list-style-type: none"><li>• <b>Colaboração ineficaz devido ao modo de colaboração inadequada.</b></li><li>• <b>Atrasos e problemas em cooperação devido à forma de trabalhar.</b></li></ul>	<ul style="list-style-type: none"><li>• Formação para os vários modos de colaboração.</li><li>• Comunicação sobre a estratégia de colaboração.</li></ul>

### Condições para a colaboração

Este tópico apresenta as práticas sugeridas a serem introduzidas desde do início do projeto para assegurar as boas condições, por exemplo, a identificação de diferenças culturais.

<b>Problemática</b>	<b>Solução</b>
<ul style="list-style-type: none"><li>• <b>Atrasos e esforço desperdiçado devido à inadequada organização do projeto ou práticas e ferramentas de colaboração em falta.</b></li><li>• <b>Falta de visibilidade do trabalho dos parceiros.</b></li></ul>	<ul style="list-style-type: none"><li>• Motivar as pessoas para a colaboração, como por exemplo, as recompensas.</li><li>• Analisar e preparar-se para as diferenças culturais entre os parceiros.</li><li>• Práticas para conhecerem uns aos outros e para se acostumarem-se a trabalhar como uma equipa.</li></ul>

### Gestão de fornecedores

Este tópico descreve que os fornecedores devem estar relacionados em vários pontos do projeto e a atenção para a deteção de potenciais problemas desde do início

<b>Problemática</b>	<b>Solução</b>
<ul style="list-style-type: none"><li>• <b>Lacunas ou a duplicação de trabalho causado por atribuições pouco claras, os atrasos causados por dependências não gerenciados.</b></li></ul>	<ul style="list-style-type: none"><li>• Comunicação sobre os papéis dos parceiros no projeto, e sobre o nível adequado de confidencialidade.</li><li>• Encontros cara-a-cara.</li></ul>

- **Entregas inesperados (conteúdo ou tempo).**
- **Requisitos mal compreendidos.**

- Envolvimento do fornecedor na definição de requisitos para garantir a disponibilidade dos requisitos de competência no *site* do fornecedor.

### Gestão de projetos

Este tópico apresenta a descrição das metas de ponto de vista GDD para acompanhar o andamento do projeto e as orientações para reagir aos desvios ao atingir essas metas.

<b>Problemática</b>	<b>Solução</b>
<ul style="list-style-type: none"> <li>• <b>Estado de projeto não bem especificado.</b></li> <li>• <b>Os atrasos causados pela falta de clareza, autoridades e prática</b></li> <li>• <b>Atrasos na tomada de decisão causadas por equipas desalinhadas</b></li> <li>• <b>Resultados do trabalho não correspondentes, causados por objetivos incompreendidos.</b></li> <li>• <b>Recursos indisponíveis.</b></li> </ul>	<ul style="list-style-type: none"> <li>• Identificar as competências requeridas e estabelecer um plano para fechar eventuais lacunas.</li> <li>• Metas do projeto a comunicar.</li> <li>• Identificar competências críticas e criação de cópias de segurança</li> <li>• Assegurar a disponibilidade de conhecimento dos objetivos do projeto.</li> </ul>

- **Práticas de engenharia**

Há uma série de soluções disponíveis especialmente para as práticas de engenharia. Em seguida algumas soluções apresentadas.

### Engenharia de requisitos

Este tópico descreve as práticas para a identificação dos requisitos ambíguos ou as diferenças na compreensão das exigências entre parceiros.

Problemática	Solução
<ul style="list-style-type: none"> <li>• <b>Dificuldades em priorizar os requisitos devido a vários pontos de vista.</b></li> <li>• <b>Documentação de exigência não clara.</b></li> <li>• <b>Dificuldades na criação e gestão de rastreabilidade.</b></li> <li>• <b>Requisitos mal compreendidos causado pelas entregas inconvenientes.</b></li> <li>• <b>Problemas causados por mudanças não comunicadas aos requisitos</b></li> <li>• <b>As inconsistências entre requisitos e outros produtos de trabalho.</b></li> </ul>	<ul style="list-style-type: none"> <li>• Formação sobre as práticas da engenharia de requisitos.</li> <li>• Assegurar que as pessoas com a melhor competência definam os requisitos.</li> <li>• Comunicação dos requisitos definidos.</li> <li>• Nível certo e o envolvimento suficiente de todas as partes na validação de requisitos de sistema produzidas.</li> <li>• Assegurar a competência dos requisitos e estrutura do produto</li> </ul>

### Arquitetura

Nesse tópico as práticas para assegurar um entendimento comum da arquitetura entre os parceiros, por exemplo, através da utilização de várias visões de arquitetura e princípios de *design*.

Problemática	Solução
<ul style="list-style-type: none"> <li>• <b>Arquitetura inadequada (não apoiar o desenvolvimento distribuído)</b></li> <li>• <b>Não claro, não comunicada.</b></li> </ul>	<ul style="list-style-type: none"> <li>• Usando visões de arquitetura para facilitar a compreensão da arquitetura.</li> <li>• Formação de efeitos de GDD sobre a arquitetura.</li> <li>• Desenvolvimento da arquitetura pelas pessoas certas.</li> <li>• Assegurar a disponibilidade de arquitetura perícia em todos os <i>sites</i>.</li> </ul>

### Design e código

Neste tópico, apresento a descrição das regras comuns de *design* e implementação que devem ser seguidas pelos parceiros, a fim de garantir uma qualidade comum e compreensibilidade do trabalho feito por outros; por exemplo os padrões de desenho a serem utilizados.

Problemática	Solução
--------------	---------

<ul style="list-style-type: none"> <li>• Dificuldades de comunicação lógica causando erros em decisões de <i>design</i>.</li> <li>• Estilos de programação diferentes e normas que afetam a compreensibilidade do <i>design</i> e código por outros.</li> </ul>	<ul style="list-style-type: none"> <li>• Comunicação e formação da arquitetura, fundamentos e princípios de <i>design</i>.</li> </ul>
---	---

### Integração

Neste tópico apresento a descrição das diferentes estratégias de integração, que ajudam a escolher o caminho mais certo para o projeto.

Problemática	Solução
<ul style="list-style-type: none"> <li>• Os atrasos nas respostas para os problemas encontrados na integração, indisponibilidade de informações.</li> <li>• Problemas na integração causados pelas entregas não sincronizadas.</li> </ul>	<ul style="list-style-type: none"> <li>• Assegurar a competência e a disponibilização de desenvolvedores para resolver os problemas encontrados na integração.</li> <li>• Criar consciência das responsabilidades de integração.</li> <li>• Disponibilidade de conhecimentos e de arquitetura em todos os <i>sites</i> partilhados.</li> </ul>

### Testes

Neste tópico apresento as práticas para apoiar a partilha de informações de teste (casos de teste, os resultados do teste, dados de teste) entre os membros das equipas.

Problemática	Solução
<ul style="list-style-type: none"> <li>• Dificuldades na replicação dos defeitos encontrados nos testes.</li> <li>• Falta de visibilidade dos testes.</li> <li>• Testes inadequados (devido à falta de conhecimento).</li> <li>• Teste ineficaz causada por partos não sincronizadas.</li> </ul>	<ul style="list-style-type: none"> <li>• Assegurar a disponibilidade de conhecimentos dos testes</li> <li>• Criação de uma atitude positiva para os testes (avaliação não pessoal).</li> <li>• Conhecimento do produto no local do teste.</li> </ul>

- **Práticas de suporte**

As práticas de suporte devem ser ajustadas para apoiar o fenómeno GDD. As melhores práticas devem ser registadas e usadas entre os parceiros. Além disso, em relacionamentos a longo prazo é útil trabalho processos de melhoria. De seguida encontra se uma amostra de soluções para apoiar as práticas.

#### Gestão de configuração

Este tópico descreve as práticas de gestão de configuração que devem ser comuns entre todos os parceiros.

Problemática	Solução
<ul style="list-style-type: none"> <li>• <b>Práticas de gestão de configuração complexa.</b></li> <li>• <b>Dificuldades em encontrar versões corretas</b></li> <li>• <b>Uso de versões erradas causando problemas em compilações.</b></li> <li>• <b>Estado do trabalho de desenvolvimento não bem especificado.</b></li> </ul>	<ul style="list-style-type: none"> <li>• Assegurar a disponibilidade de competência de gestão de configuração.</li> <li>• Formação sobre as práticas de gestão de configuração.</li> </ul>

#### A garantia de qualidade

Este tópico descreve as práticas de gestão de qualidade que devem ser comuns entre todos os parceiros.

Problemática	Solução
<ul style="list-style-type: none"> <li>• <b>Práticas diferentes e desconhecidas dos parceiros.</b></li> <li>• <b>Responsabilidades desconhecidas para a garantia da qualidade.</b></li> <li>• <b>As diferenças de qualidade do trabalho entre os parceiros.</b></li> </ul>	<ul style="list-style-type: none"> <li>• Motivação para a garantia de qualidade ao longo dos <i>sites</i>.</li> <li>• Comunicação de práticas.</li> <li>• Assegurar a disponibilidade de competência dos que garantem a qualidade.</li> </ul>

### A partilha de informação

Este tópico descreve as práticas de documentação que tornam o compartilhamento de informações em projetos distribuídos mais fácil (compreensibilidade e disponibilidade de documentação).

<b>Problemática</b>	<b>Solução</b>
<ul style="list-style-type: none"><li>• <b>Dificuldades em saber quem tem a informação relevante e a que é necessária.</b></li><li>• <b>As dificuldades em definir os meios adequados para a partilha de informação.</b></li><li>• <b>Dificuldades em saber que informação deve ser comunicada e como (o que já é conhecido e o que não é).</b></li><li>• <b>Responsabilidades pouco claras para o compartilhamento de informações.</b></li></ul>	<ul style="list-style-type: none"><li>• <b>Criar consciência da importância da partilha de informação e a motivação para fazê-lo.</b></li><li>• <b>Habilidades comunicativas de líderes de projeto.</b></li></ul>

### Infraestrutura

Este tópico apresenta a lista de verificação que ajuda a garantir que a infra-estrutura é adequada para qualquer situação em questão GDD.

<b>Problemática</b>	<b>Solução</b>
<ul style="list-style-type: none"><li>• <b>Diferentes ferramentas incompatíveis utilizados pelos parceiros.</b></li><li>• <b>Problemas na disponibilidade de ferramentas</b></li><li>• <b>Diferentes maneiras de usar as ferramentas.</b></li><li>• <b>Incapacidade de usar ferramentas de comunicação de forma eficaz.</b></li></ul>	<ul style="list-style-type: none"><li>• <b>Formação para a utilização da infraestrutura.</b></li><li>• <b>Disponibilidade de competência para a resolução de problemas na infraestrutura.</b></li><li>• <b>Ferramentas de formação</b></li><li>• <b>Reuniões virtuais.</b></li></ul>

### Gestão de competências

Estes tópicos apresentam as práticas de desenvolvimento da equipa a considerar quando a construímos, por exemplo, as competências adequadas para criar uma atmosfera amigável.

Problemática	Solução
<ul style="list-style-type: none"><li>• <b>O desconhecimento das competências disponíveis</b></li><li>• <b>Utilização não otimizada dos recursos.</b></li></ul>	<ul style="list-style-type: none"><li>• Criar consciência de competências e motivação para envolver as pessoas não conhecidas com as competências certas para as tarefas.</li></ul>

### Melhoria contínua

Esse tópico apresenta as métricas e práticas de medição que fornecem informações sobre a eficácia da colaboração durante e depois do projecto.

Problematica	Solucao
<ul style="list-style-type: none"><li>• <b>Práticas diferentes e desconhecidas de parceiros, as responsabilidades pouco claras.</b></li><li>• <b>Informações não partilhadas sobre as lições aprendidas e as melhores práticas.</b></li></ul>	<ul style="list-style-type: none"><li>• Formação para as práticas utilizadas.</li><li>• Criação de motivação para partilhar problemas</li><li>• Melhores práticas e lições aprendidas.</li></ul>

**Mito:** #2

Mito: As diferenças culturais são uma desvantagem.

Credibilidade: Alta

Realidade: As empresas que querem aceder ao maior sucesso possível ao maior aproveitamento e ao poder da inovação terão de trabalhar intensamente para criar culturas de inovação e a capacidade de ficar acima dos seus concorrentes. Sem a existência de uma só cultura, sem a incentivo de pensar "fora da caixa", não há oportunidades para competir no mundo dos negócios.

Como nos é mostrado por Hofstede [47] (2001), as pessoas com diferentes origens culturais se comportam-se de uma forma diferente. Sugestões que podem ajudar a superar este problema são: 1) estabelecer uma pasta que lista todos os profissionais envolvidos no desenvolvimento de certas partes da arquitetura e 2) iniciar todos os projetos com uma reunião. Estabelecer contato é mais fácil quando todos estão localizadas no mesmo sítio. Outro impacto potencialmente positivo da distribuição global é a inovação, como a mistura de desenvolvedores com diferentes origens culturais podem desencadear novas ideias [10] [14] [12].

O impacto das culturas na gestão das organizações tem sido um alvo constantemente falado.

O Hofstede [47], um dos grandes estudiosos do tema, argumenta que a globalização dos negócios não leva necessariamente a uma só cultura. Em outras palavras, as soluções de gestão, para ser bem-sucedidas, devem levar em conta as diferenças culturais que regem as sociedades humanas. Segundo este autor, a cultura funciona como uma programação mental constituída por padrões de pensamento, sentimentos e potenciais ações que cada indivíduo carrega. Isto é resultado da aprendizagem contínua e predetermina parcialmente o comportamento dos seres humanos. Apesar da enorme diversidade de mentes, há uma estrutura que serve como o ponto de partida, ponto para a compreensão mútua e das diferenças culturais.

Se a cooperação e a coordenação das equipas de projeto é um desafio global, então a questão cultural também é identificada como a causa de falhas nos projetos desta natureza. Porque cada indivíduo transporta a sua cultura (crenças e valores) e o chefe de projeto precisa de estar consciente dos impactos da diversidade cultural sobre o desempenho dos projetos. No entanto, apesar de tal importância, a pesquisa mostra que o assunto tem sido pouco estudado por autores no campo da gestão de projetos [2] [48].

Do ponto de vista operacional, o papel do gestor é focado em questões funcionais tais como recrutamento, seleção, desenvolvimento, avaliação e compensação, explorando assim o impacto da gestão de pessoas e a complexidade de operar a internalização.

Este raciocínio também se aplica com as políticas de gestão de pessoas, que deve ser flexível, capaz de lidar com as diferenças, o que indica uma grande necessidade de observação das diferenças demográficas e culturais-

Em relação ao desempenho das equipas, não há resultados que demonstram que o desempenho pode sofrer impactos negativos em ambientes de projetos internacionais quando comparados com projetos locais, especialmente no início. O desempenho é melhorado e pode ser ainda maior quando os membros dessas equipas multiculturais se acostumaram a colaborar e a desenvolver estratégias para superar as barreiras culturais e linguísticas [45]. O Nurick e o Thamhain [49] oferecem uma série de recomendações para o processo de desenvolvimento de equipas globais em todo o ciclo de vida do projeto. No início do projeto, o gestor deve investigar o envolvimento do pessoal a fim de criar um ambiente favorável para o desempenho do mesmo, em sum todos estarem comprometidos com o resultado a ser alcançado. A definição adequada das interfaces, responsabilidades com os deveres, relações de autoridade, comunicação, canais e protocolos de trabalho devem ser desenvolvidos e a informação deve ser partilhada. Na atribuição e organização das equipas é importante que a liderança seja cuidadosamente definida no início do projeto, como a credibilidade de líder com a equipa, com a alta administração e as partes interessadas é crucial para a gestão e integração de diferentes tarefas que são necessários para atingir o projeto. A construção de uma imagem do projeto de alto desempenho é fundamental e exige o suporte organizacional, assim como a sua visibilidade do projeto e sistema de recompensa para os membros da equipa para assim atrair profissionais qualificados.

**Mito: #3**

Mito: A identificação dos riscos é feita logo na primeira fase do projeto tal como o impacto, a probabilidade de ocorrência e a prioridade de tratamento.

Credibilidade: Baixa

Realidade: Recordando a definição de projeto presente no PMBOK: [50] “Um projeto é um esforço temporário empreendido para criar um produto, serviço ou resultado exclusivo.” A organização do trabalho em projetos, sejam eles de desenvolvimento de *software* ou não, se dá-se fundamentalmente para que seja possível lidar com os desafios inerentes à criação de um resultado novo, exclusivo e diferente em algum aspeto de tudo que já foi criado anteriormente. Como há sempre algum grau de novidade, as incertezas são inerentes ao

conceito de projeto e é

fundamental que o gerenciamento dos riscos ocorra em algum nível. A única certeza é que os riscos existem e se os mesmos não forem devidamente tratados, a probabilidade de que algum deles ocorra e prejudique algum aspecto do projeto é de aproximadamente 100%.

Os riscos gerais envolvidos em GDD, em prática tem a ver com a comunicação entre os parceiros; por exemplo, um problema/solução não divulgado pode ser um problema nas relações cliente-fornecedor. Além disso, as atribuições claras, falta de confiança entre os parceiros, dificuldades em chegar a um acordo sobre os direitos de propriedade intelectual e a falta de fiabilidade do cronograma de desenvolvimento dos parceiros, foram vistos como fatores de risco para qualquer modo de colaboração. Por outro lado, do ponto de vista do cliente, a qualidade do produto adquirido (por exemplo, a fiabilidade e o desempenho) torna-se muito arriscado. Finalmente, as questões de competência, tais como a competência de novos parceiros e um enfraquecimento da própria competência da pessoa, também foram mencionados como riscos. Estes riscos são semelhantes aos que outros autores puseram em evidência. Por exemplo, o Ebert [10] Identificou uma lista de riscos para projetos de GDD na última década em uma multiplicidade de projetos e situações que abrangem quatro continentes. Eles não são específicos para determinados sectores ou portes de empresa, mas sim para os processos do ciclo de vida subjacentes e práticas de gestão.

Como no desenvolvimento de *software* tradicional, a tarefa possui fatores de risco em projetos de DSD, porém com razões ligeiramente diferentes. Quando o projeto é dividido e distribuído em distintos locais dúvidas nos requisitos funcionais podem surgir devido à falta de formalidade e transparência ou seu propósito [28] [51]. Com a distribuição das tarefas, também pode ocorrer o aumento de comunicação e integração entre os locais de distribuição devido ao acoplamento das tarefas [52].

Os fatores de risco também estão relacionados com a estrutura que agrupa dimensões temporais, geográficas e colaborativas em ambientes globais. A distância geográfica complica o monitoramento do progresso das equipas, enfraquece relações sociais, além de aumentar os custos com viagens e restringir contatos cara-a-cara [28] [20]. A distância temporal aumenta a complexidade de planeamento e causa improdutividades e atrasos em respostas, além de complicar as configurações de horas [30] [45].

Na distribuição geográfica, diversos problemas podem surgir uma vez que os *stakeholders* não utilizam necessariamente a mesma língua e cultura organizacional [52]. A falta de interação entre os participantes cara-a-cara, também podem gerar problemas na transferência do conhecimento adquirido, experiência no domínio do projeto.

- **Tarefa**

A tarefa corresponde às atividades a fim de uma organização ou às operações que levam à produção de bens e serviços. O termo Tarefa pode ser descrito como uma razão de viver da organização. No desenvolvimento de *software* uma tarefa é normalmente definida através dos artefatos e funcionalidades do processo, i.e. uma tarefa de desenvolvimento diz “O QUE” o software deve realizar e “COMO”. Várias tarefas relacionadas a funcionalidades que aumentam a exposição ao risco têm sido identificadas: tamanho da tarefa, complexidade, incerteza da tarefa, a estabilidade da tarefa, a ambiguidade e a existência da descrição da tarefa.

<b>Fatores de Riscos</b>	<b>Indicação de Risco Baixo</b>	<b>Indicação de Risco Médio</b>	<b>Indicação de Risco Alto</b>
<b>Formalidade e Transparência das tarefas</b>	As atividades de cada local estão bem definidas.	Alguns locais necessitam de ter as atividades explicitamente definidas.	As atividades dos locais estão sobrepostos com quase nenhuma organização modularizada.
<b>Tratamento dos Requisitos Não-Funcionais</b>	As restrições de segurança, desempenho e disponibilidade são conhecidas e capturadas de forma efetiva.	A maioria das restrições são conhecidas, porém algumas precisam de uma maior atenção.	Existem restrições de segurança, desempenho e disponibilidade que impactam diretamente o projeto em desenvolvimento.
<b>Dúvida nos requisitos funcionais</b>	A equipa entende a funcionalidade e não tem dúvida na tarefa e ser executada.	A equipa possui dúvidas isoladas na tarefa a ser executada.	A equipa não consegue compreender bem a tarefa a ser executada.

<b>Acoplamento das tarefas</b>	Quase não existe nenhuma necessidade de coordenação entre os locais de desenvolvimento.	Existe uma certa necessidade de coordenação entre os locais de desenvolvimento.	Existe uma necessidade significativa para coordenar os locais de desenvolvimento.
<b>Criticidade da tarefa</b>	As falhas na tarefa são toleradas, permitindo o seu conserto.	Algumas falhas podem impactar momentaneamente o projeto mas não encerram o projeto.	As falhas encontradas ameaça o sucesso do projeto.
<b>Complexidade da tarefa</b>	Não há grande necessidade de documentação da tarefa por ser de fácil compreensão.	É recomendado documentar a funcionalidade para cada parte a ser desenvolvida.	A tarefa em si possui um nível de detalhamento que precisa se documentado para que se possa entender os seus relacionamentos.
<b>Estabilidade dos requisitos</b>	Os requisitos raramente mudam.	Os requisitos eventualmente mudam.	Os requisitos mudam frequentemente.

- **Estrutura**

A estrutura está ligada aos processos organizacionais, aos sistemas de comunicação organizacionais e ao fluxo dos processos de trabalho. Por Estrutura, denota-se sistemas de comunicação e sistemas de fluxo de trabalho. Embora não torne explícito, a estrutura da dimensão padronizada, *i.e.* valores e normas e a dimensão comportamental, *i.e.* padrões reais de comportamento conforme se os atores se comunicam, exercem autoridade ou trabalham.

<b>Distância temporal</b>	Os fusos horários não causam nenhum problema de coordenação.	Os fusos horários requerem um tratamento restrito para algumas localidades.	Existem muitos fusos-horários que impactam negativamente o desenvolvimento do projeto.
---------------------------	--	---	--

<b>Diferença nas leis</b>	As leis são similares.	As leis diferem um pouco, mas o projeto pode ser desenvolvido sem problemas.	As leis possuem diferenças que podem causar a finalização do projeto.
<b>Distância geográfica</b>	Distância entre os locais não significativa para causar problemas.	Distância entre os locais é grande, porém os impactos se restringem-se a casos pequenos e isolados.	Distância significativa entre os locais causando diversos problemas, por exemplo problemas de comunicação e custos de locomoção.
<b>Divergências no processo de desenvolvimento</b>	Os processos nas diferentes localidades não variam, uma vez que todos utilizam o mesmo padrão.	Os processos variam, mas possuem um padrão parcialmente comum.	Os processos são diferentes em todas as localidades envolvidas.
<b>Maturidade do processo</b>	O nível de maturidade do processo entre os locais é igual aos trabalhos alinhados aos objetivos e negócio da empresa.	O nível de maturidade dos processos nos locais distribuídos é semelhante com pouca variação.	Existem diferentes maturidades do processo e inconsistência no trabalho realizado e os diversos locais distribuídos.
<b>Quantidade de locais de distribuição</b>	O projeto está distribuído em poucos locais.	O projeto está distribuído em alguns locais.	O projeto está distribuído em vários locais.

- **Ator**

Os atores são todas as pessoas que estão envolvidas na realização das tarefas organizacionais. Eles representam indivíduos ou grupos de *stakeholders* que podem promover pedidos ou beneficiar do desenvolvimento do *software*. Os atores incluem clientes, gerentes, mantenedores, desenvolvedores e usuários. Exemplos fatores de riscos de *software* relacionados a atores são: deficiências pessoais, falta de compromisso e conhecimento, diferenças entre *stakeholders*,

expectativas erradas, crenças erradas, profissionais não éticos, troca de funcionários, políticas pessoais e oportunismo.

<b>Diferenças socioculturais</b>	Ética e normas similares nos locais de desenvolvimento.	As éticas e normas possuem algumas variações nos locais de desenvolvimento.	As éticas e normas são divergentes nos locais de desenvolvimento.
<b>Diferenças culturais de trabalho</b>	As pessoas compartilham e entendem as informações nos locais distribuídos.	As pessoas conhecem as diferenças entre os locais distribuídos e se comunicam sem problemas significantes.	As pessoas não conhecem as diferenças culturais e possuem um diferente entendimento de conceitos.
<b>Diferenças de língua</b>	As pessoas falam a mesma língua e possuem regras de comunicação iguais.	As pessoas utilizam uma língua comum e possuem regras de comunicação similares.	As pessoas comunicam em línguas diferentes entre si e com regras de comunicação distintas.
<b>Distância de conhecimento</b>	Não existem lacunas de conhecimento entre os locais distribuídos.	Algumas especificações são criadas com erros devido à falta de conhecimento específico.	Existem grandes problemas relacionados com as habilidades necessárias ao desenvolvimento do projeto nos locais distribuídos.
<b>Transferência do conhecimento</b>	Os conhecimentos gerados são compartilhados e integrados nos locais distribuídos.	Existem conhecimentos pontuais que não são compartilhados e integrados.	Os conhecimentos ficam isolados em cada local distribuído.
<b>Relacionamento</b>	As pessoas	As pessoas por vezes	As pessoas não se

<b>pessoal</b>	relacionam-se pessoalmente.	relacionam-se pessoalmente.	relacionam pessoalmente.
<b>Motivação da equipe</b>	A equipa está motivada para trabalhar de forma distribuída.	Parte da equipa está motivada a trabalhar num ambiente distribuído.	A equipa não está motivada a interagir neste tipo de ambiente.
<b>Experiência no domínio do projeto</b>	A equipa tem experiência anterior no domínio do projeto.	A equipa possui parte do conhecimento necessário ao desenvolvimento das funcionalidades.	O domínio do projeto é totalmente novo para os participantes envolvidos.
<b>Falta de confiança</b>	Existe sinergia e relacionamento físico entre as equipas.	Existe um certo relacionamento físico entre as equipas distribuídas.	Existe rivalidade entre as equipas distribuídas e receio de contatos.

- **Tecnologia**

A tecnologia está ligada ao conjunto de elementos capazes de resolver os problemas na organização de forma direta. Por exemplo, a técnicas de mensuração da produtividade, sistemas computacionais e computadores. De acordo com o conceito de “invenções para resolução de problemas” foram incluídos métodos tecnológicos, ferramentas e infraestrutura para desenvolver e programar sistemas de *software*. Estas tecnologias podem criar riscos consideráveis especialmente se elas são não confiáveis, ineficientes, não-padronizadas, incompatíveis ou têm limitações funcionais.

<b>Inadequação das tecnologias de segurança</b>	Os dados do projeto são inteligíveis apenas para os membros envolvidos no projeto protegidos por restrições de acesso.	Existem algumas restrições de acesso às informações para os participantes do projeto.	Os dados dos projetos podem ser compreendidos por qualquer pessoa interessada havendo falta de padrão nas restrições de acesso.
<b>Incompatibilidade tecnológica</b>	Todas as localidades utilizam o mesmo <i>hardware</i> e software no desenvolvimento do projeto.	Existem algumas divergências entre as ferramentas de desenvolvimento do projeto.	Cada localidade usa a ferramenta da sua escolha para o desenvolvimento do projeto.
<b>Adequação das ferramentas de apoio</b>	As ferramentas de apoio utilizadas no desenvolvimento correspondente às expectativas de utilização em projetos distribuídos.	Algumas exceções não correspondem às expectativas de utilização em projetos distribuídos.	Ferramentas chaves para apoio no desenvolvimento do projeto estão aquém do desempenho esperado.

Tabela 4. Fatores de Riscos

## 4.2 Outros Mitos

Apesar de muitos aspetos serem estudados, a percepção de uns tópicos estão longe de serem verdade. Ainda temos de ter em consideração a presença de debates sobre as práticas globais e o peso da credibilidade. De seguida uma apresentação de mitos secundários, menos ponderado e a exploração do meu critério de credibilidade.

### **Mito: #4**

Mito: É uma solução rápida.

Credibilidade: Intermédia

Realidade: Uma solução rápida através de GDD criador a longo prazo. Entre os motivos para que isto ocorra, demanda-se um maior controlo, maior flexibilidade, menores preços a longo prazo além de manter a cultura interna da empresa, pedindo-se uma maior complexidade de operacionalização e mais tempo [48].

### **Mito: #5**

Mito: Não é seguro.

Credibilidade: Alta

Realidade: A globalização pode e deve melhorar a segurança dos seus dados. Porque qualquer violação de dados pode colocar alguém fora do negócio. A segurança a nível de aplicativo e infraestrutura de transferência, de dados com controlos que igualam ou superam a proteção de defesas do *Firewall*.

### **Mito: #6**

Mito: GDD é uma competência essencial para o negócio.

Credibilidade: Alta

Realidade: Só porque o processamento tem mais poder sobre a categoria "financiamento", não significa que seja fundamental para o seu negócio. Uma competência central fornece benefícios para o cliente, é difícil para os concorrentes imitar, e pode ser aproveitado amplamente em muitos produtos e mercados e assim os fornecedores podem, portanto, aumentar a sua vantagem competitiva, fornecendo aos clientes benefícios para o processo.

**Mito: #7**

Mito: GDD só é útil para reduzir o número do pessoal.

Credibilidade: Alta

Realidade: Se a redução de pessoal é de facto um objetivo, o desenvolvimento global de *software* tem sido apresentado obter esse objetivo sem perda de produtividade ou capacidade. No entanto, uma visão de longo prazo está a tornar-se cada vez mais importante. Além disso, a eficiência adquirida é mais rápida e o processamento menos caro o que pode aumentar a competitividade global da empresa e o que beneficia todos os funcionários.

**Mito: #8**

Mito: Trabalhar em paralelo apresenta um resultado longínquo do fim do prazo e do orçamento.

Credibilidade: Intermédia

Realidade: Em particular, a harmonia promove confiança entre a central e a equipa deslocada, e essa confiança prossegue em sincronização de trabalho. No entanto, na prática sente-se a presença constante das barreiras e problemas, mas não impede atingir os objetivos comuns.

**Mito: #9**

Mito: Não conhecer a equipa deslocada não afeta o processo e trabalho.

Credibilidade: Fraca

Realidade: O conhecimento do membro da equipa que se trabalha é um aspeto profundamente importante para trabalhar em equipas globais. As práticas de trabalho, a consciência das diferenças culturais, bem como a estrutura organizacional, pode ser definida como conhecimento obrigatório de cada membro da equipa. Outro aspeto a levar em conta está em processos de mudança organizacional. Quando uma nova estratégia de negócios é implementada, parte do conhecimento existente precisa de ser mudado.

**Mito: #10**

Mito: **A central perde o controlo sobre os seus dados.**

Credibilidade: Intermédia

Realidade: Embora às vezes pareça que não há controlo sobre os dados, há uma série de ferramentas que os ajudam a controlar melhor os dados.

Para Herbsleb [1], as ferramentas de apoio ao desenvolvimento de software podem suprir a necessidade de comunicação dos projetos globais, possibilitando o controlo de versões, o histórico de alterações, os artefatos e a documentos do projeto em geral.

**Mito: #11**

Mito: Somos muito pequenos.

Credibilidade: Baixa

Realidade: A verdade é que funciona bem para as pequenas empresas, porque as economias de escala podem fornecer recursos e benefícios para as empresas a um custo muito menor. Além disso, o que representa os funcionários e gerentes financeiros em empresas menores, muitas vezes têm responsabilidades variadas, e a globalização como função tática permite que eles concentrem os esforços em tarefas mais estratégicas.

**Mito: #12**

Mito: Somos muito grandes.

Credibilidade: Baixa

Realidade: A infraestrutura do mercado do desenvolvimento de *software* global aumentou tremendamente, trazendo mais capacidade de lidar com organizações de todos os tamanhos. Além disso, as grandes empresas podem usar o GDD para apoiar uma linha específica de negócios ou localização geográfica, ou para limpar uma área problemática. Também é uma opção valiosa para uma aquisição e pode lidar com o processamento da empresa adquirida, quer até que as operações estejam consolidadas ou como uma estratégia de longo prazo.

**Mito: #13**

Mito: É impossível construir um *software* de qualidade máxima.

Credibilidade: Intermédia

Realidade: A natureza transacional do desenvolvimento de *software* tem levado ao surgimento de dois mitos completamente opostas: ou o *software* produzido é de boa qualidade, ou, em alternativa, o *software* produzido é de má qualidade.

Em prática, a maioria das empresas de terceirização estão em dívida com testes de aceitação de clientes contratuais de *software*. Se os testes de aceitação não são uma parte de um contrato de desenvolvimento, é pouco provável que o resultado seja um *software* que pode implantar com pouca confiança. A qualidade do produto requer esforço concentrado.

**Mito: #14**

Mito: O GDD é sempre uma opção para poupar dinheiro.

Credibilidade: Alta

Realidade: Um dos objetivos é maximizar os lucros, mas não se consegue introduzir uma nova tecnologia sem uma taxa de excesso. Há sempre uma possibilidade da transferência das competências para obter novos negócios, o que cria um aumento dos custos.

**Mito: #15**

Mito: A redução de custos é a única vantagem.

Credibilidade: Alta

Realidade: A mensagem principal é que não se deve concentrar só em recursos, mas e nas competências e nas atividades estratégicas. Em média as empresas apenas consideraram os benefícios de custos a curto prazo de GDD, em vez de diminuir os investimentos de capital de longo prazo através dessa estratégia.

**Mito: #16**

Mito: Os países em desenvolvimento ainda não estão prontos para obter benefícios GDD.

Credibilidade: Baixa

Realidade: A revolução dos dados e os benefícios são um fenómeno global. Alguns dos

benefícios mais importantes e algumas das maiores oportunidades encontram-se no mundo em desenvolvimento, onde muitas vezes as tecnologias são obsoletas.

GDD oferece uma série de vantagens, permite que as organizações beneficiem do acesso a recursos qualificados. De acordo com o inventário industrial, uma das razões mais comuns para a colaboração é adquirir competências em tecnologia. Outras razões incluídas são a criação de novas oportunidades de negócios com novos parceiros, a flexibilidade no que diz respeito ao número de recursos internos e superar problemas de disponibilidade de recursos internos.

**Mito: #17**

Mito: O GDD gera ansiedade e caos, que são prejudiciais à empresa.

Credibilidade: Intermédia

Realidade: A verdade é que se cria ansiedade mas não precisa de ser prejudicial e o caos não precisa de ser o resultado. Quando o processo é adequadamente administrado e implementado, é possível evitar cicatrizes profundas.

## 4.3 Mitos sobre os processos de modelagem

O processo de desenvolvimento e entrega de *software* na realidade é muito mais complexo do que imaginamos, pois tem que responder os requisitos iniciais e estarem preparados para qualquer alteração de local/diferença de fuso horário. E uma das propostas para este tipo de problema são as metodologias não tradicionais que validam aumentar a eficácia da equipa de um projeto de desenvolvimento. De seguida, os mitos mais encontradas no ambiente dos processos de modelagem.

**Mito: #18**

Mito: A competência das pessoas é desconsiderada se implementar um bom modelo de gestão.

Credibilidade: Baixa

Realidade: Problemas provenientes dessas causas incluem, por exemplo, a utilização ineficaz dos recursos como competências, obstáculos na resolução de problemas aparentemente pequenos de trabalho com falhas devido à falta de competência ou informação de fundo. Essas causas também podem levar a uma falta de transparência nos outros partidos de trabalho, atribuições mal compreendidas e, assim, as entregas de partes defeituosas, os atrasos causados pela espera para os outros partidos de entrada e duplicar o trabalho ou áreas descobertas.

Os processos são de facto bastante valorizados nos modelos. Mas há também resultados / práticas / objetivos específicos referenciando diretamente a necessidade da competência e conhecimento das pessoas em relação às atividades que executam.

**Mito: #19**

Mito: Scrum é o mais praticado porque usa uma abordagem iterativa e não cascata.

Credibilidade: Intermédia

Realidade: Não há uma definição de uso de um ciclo de vida de um projeto específico. O que é necessário para que haja aderência aos modelos é a definição de qual o ciclo de vida será utilizado para cada um dos projetos. Pode ser cascata, iterativo, misto, incremental, entrega evolutiva ou qualquer outro.

Os métodos ágeis, como Scrum, são cada vez mais aplicados em GDD para alcançar os benefícios de ambos os métodos. Métodos ágeis prometem benefícios de lidar com as mudanças de requisitos durante todo o ciclo de desenvolvimento, extensa colaboração entre clientes e desenvolvedores e entrega antecipada e frequente de *software* [53]. No entanto, os métodos ágeis normalmente assumem um contexto de projeto que permite a interação estreita entre os membros da equipa. Com efeito, um dos princípios do Agile Manifesto, a conversa cara-a-cara é o método mais eficiente e eficaz de partilha de informação em uma equipa de desenvolvimento [54]. Este apresenta um problema significativo em GDD. As equipas de desenvolvimento e / ou seus membros podem ser espalhados ao redor do globo. A superação dos desafios temporais, geográficos e socioculturais baseadas na distância em tal contexto define os obstáculos significativos para superar e perceber os benefícios de métodos ágeis em GDD.

Scrum por ser uma abordagem interativa, ganha uma melhor visibilidade das atividades do projeto e uma melhor colaboração das partes interessadas, confiança, conscientização da equipa e compreensão compartilhada.

**Mito: #20**

Mito: O desenvolvimento de *software* distribuído irá gerar uma qualidade inferior independentemente do modelo do protótipo implementado.

Credibilidade: Intermédia

Realidade: As práticas de *Extreme Programming*, como a integração contínua e desenvolvimento orientado a testes, garante um nível alto de qualidade.

As práticas ágeis promovem uma comunicação mais direta e frequente entre os membros das equipas distribuídas. Em teoria, a comunicação é o desafio em GDD que mais se beneficia com a adoção de um processo de desenvolvimento de *software* que segue os princípios da metodologia XP [55] [56] [57] [58].

A liberdade de expressão promovida pelo processo ágil faz com que os membros da equipa se sintam mais a vontade em estabelecer contatos com as equipas distribuídas. Estas praticam ajudam-na a equipa a desenvolver uma intensa comunicação para além de manter uma forte comunicação interna entre os seus membros.

**Mito: #21**

Mito: Não é funcional o Scrum e / ou Agile com as equipas distribuídas.

Credibilidade: Intermédia

Realidade: Em *Scrum*, é preferível que os membros da equipa fiquem juntos porque proporciona um maior grau de colaboração e promove um nível mais elevado e eficaz. No entanto, pode haver um elevado nível de eficácia num ambiente distribuído. As práticas do Scrum em GDD melhoram a comunicação, confiança, motivação e qualidade do *software* [58].

**Mito: #22**

Mito: O método Ágil não ajuda a desenvolver o processo mais rapidamente.

Credibilidade: Alta

Realidade: Agile é uma maneira muito disciplinada de entregar *software*, muitas vezes mais disciplinada que os métodos tradicionais. Se seguir os seguintes passos: testar; obter feedback; enviar regularmente *software* e informar os riscos o mais cedo possível, o resultado é um processo de desenvolvimento mais organizado e rápido.

A maioria das metodologias ágeis nada possuem de novo [59]. O que as diferencia das metodologias tradicionais são o enfoque e os valores. A ideia das metodologias ágeis é o enfoque nas pessoas e não em processos ou algoritmos. Além disso, existe a preocupação de gastar menos tempo com documentação e mais com a implementação. Por exemplo, para seguir estritamente o planejamento, pode ser necessário que a equipe trabalhe sob pressão e faça muitas horas extra, o que prejudica a qualidade do *software*.

Um artigo [60] comparando métodos ágeis com as metodologias tradicionais pesadas mostrou que os projetos usando os métodos ágeis obtiveram melhores resultados em termos de cumprimento de prazos, de custos e padrões de qualidade. Além disso, o mesmo estudo mostra que o tamanho dos projetos e das equipes que utilizam as metodologias ágeis têm crescido.

Mito: **#23**

Mito: Agile é implementado somente em projetos pequenos.

Credibilidade: Alta

Realidade: O método pode adaptar-se a qualquer tipo e tamanho de projeto. As organizações têm reportado sucesso em programas ágeis com mais de 500 pessoas; 75% das organizações que executam projetos ágeis estão a trabalhar em projetos com médio nível de complexidade e projetos maiores.

Um dos desafios é como usar essas metodologias ágeis em grandes empresas e equipes, uma vez que normalmente essas metodologias são baseadas em equipes pequenas. Neste caso, pelo menos é necessário resolver os problemas de comunicação internos na equipe, uma vez que é comum em grandes empresas que os funcionários estejam separados geograficamente. Apesar do interesse crescente no uso das metodologias ágeis, ainda faltam casos de sucesso de seu uso em projetos grandes e críticos. Quanto mais organizações usem as metodologias ágeis, melhores serão os resultados empíricos em termos de vantagens, desvantagens, riscos

e procedimentos para a adoção nas organizações. Mesmo assim, os resultados iniciais em termos de qualidade, confiança, datas de entrega e custo são promissores.

Concluindo, os métodos ágeis podem ser implementados em projetos de qualquer dimensão.

Mito: **#24**

Mito: Agile é anti documentação.

Credibilidade: Alta

Realidade: A documentação é necessária principalmente para atender o projeto. É um guia valioso para acompanhar o desenvolvimento do início até ao fim. No entanto, não nos esqueçamos que o objetivo principal de desenvolvimento de *software* é a criação de software, não de documentos.

Mito: **#25**

Mito: Se gerirmos todos os documentos que são solicitados pelo modelo, não resta tempo para o mais importante: *software* de qualidade.

Credibilidade: Intermédia

Realidade: Os modelos não obrigam a seguir nenhum documento específico. O modelo apenas exige que algumas formalizações sejam realizadas. A obtenção de compromissos por parte de clientes e fornecedores, por exemplo, é muito mais segura se for formalizada.

Mito: **#26**

Mito: Agile requer muito trabalho.

Credibilidade: Alta

Realidade: Ao contrário de um desenvolvimento tradicional, Agile reduz o trabalho, pois prevê ciclos de entregas mais curtos e coleta feedback dos usuários mais cedo.

Mito: **#27**

Mito: Agile é a bala de prata.

Credibilidade: Alta

Realidade: Agile não é a solução para tudo. Há sempre uma possibilidade de falhar num projeto Agile da mesma forma que num projeto tradicional. Ainda pode haver uma falha mais rápida usando Agile (devido à transparência e visibilidade que ela traz), mas infelizmente não é uma bala de prata ou uma desculpa para parar de se utilizar.

As abordagens ágeis são geralmente considerados eficazes para projetos com elevada incerteza [53]. Scrum é uma abordagem eficaz para a gestão de projetos com equipas pequenos [53] (Abrahamsson 2002). Sutherland e Schwaber [61] argumentam que Scrum também pode ser utilizado para equipas grandes e distribuídos.

As principais questões podem ser resumidas como as diferenças de cultura, de pessoal, de fusos horários, da comunicação e da gestão do conhecimento.

A seguinte tabela mostra as práticas gerais em vigor para abordar estas questões. Em geral, estes problemas podem ser resolvidos com sucesso por meio de processos adequados.

Desafios em Desenvolvimento distribuído		Desafios em Desenvolvimento Distribuído Ágil	Soluções/Práticas
<b>Desafios de comunicação</b>	<ul style="list-style-type: none"> <li>• Dificuldade em começar a comunicar</li> <li>• Mau entendimento</li> <li>• Diferença horária</li> <li>• Aumento do custo de comunicação e do tempo.</li> </ul>	<ul style="list-style-type: none"> <li>• Necessidade de comunicação</li> </ul>	<p>Melhorar a comunicação</p> <ul style="list-style-type: none"> <li>• Horas de trabalho sincronizadas</li> <li>• Comunicação informal através de canais formais</li> <li>• Coordenação balanceada</li> <li>• Comunicação constante</li> </ul> <p>Facilitar o conhecimento partilhado</p> <ul style="list-style-type: none"> <li>• Manter o repositório de produto/process</li> </ul>

			o
<b>Falta de controlo</b>	<ul style="list-style-type: none"> <li>Dificuldade em controlar o processo e a qualidade entre as equipas distribuídas</li> </ul>	<ul style="list-style-type: none"> <li>Requisitos de qualidade envolvida vs fixas</li> </ul>	<p>Confia mas verifica</p> <ul style="list-style-type: none"> <li>Qualidade distribuída</li> <li>Completar a comunicação informal com documentação Ajustar o processo continuamente</li> <li>Planear interações para finalizar os requisitos e desenvolver o <i>design</i></li> <li>Documentar os requisitos nos vários níveis de formalidade</li> </ul>
<b>Falta de confiança</b>	<ul style="list-style-type: none"> <li>Falta de confiança entre os membros da equipa distribuída</li> <li>Falta de moral entre os membros da equipa</li> </ul>	<ul style="list-style-type: none"> <li>Acordo formal vs informal</li> <li>Falta de coesão da equipa</li> </ul>	<p>Construir confiança</p> <ul style="list-style-type: none"> <li>Visitas frequentes por parte dos parceiros distribuídos</li> <li>Construir coesão entre a equipa</li> </ul>

Tabela 2. Desafios em desenvolvimento tradicional e Àgil

## 4.4 Mitos e Realidades vividas dentro do estágio

Durante a minha experiência e depois de viver os mitos dentro da GlobalSoftware, a minha visão teórica mudou de ângulo. Os procedimentos, os métodos, as estratégias, os passos, as soluções e muitas outras ações que no meu ponto de vista variavam da atual GDD verdade.

A principal questão sobre este mito: “Não há solução para cada específica problemática em GDD” ainda continua a ser melhor investigada em diferentes cenários. Será mesmo um mito? As soluções são relevantes e associao como uma acreditabilidade intermédia pela simples razão de esta afirmação continuar uma questão aberta e na GlobalSoftware.

Outro aspeto observado sobre este mito notado: “Não conhecer a equipa deslocada não afeta o processo e trabalho”, considero pouco verdade uma vez que criar laços, existir a interação e principalmente confiança entre as duas partes paralelas trará conhecimento geral sobre cada membro e só melhorá a colaboração entre eles.

Entretanto afirmação: “As diferenças culturais são uma desvantagem” têm duas realidades e a verdade é abstratamente considerável, porque depende das das hipóteses inicialmente indicadas. Por experiência própria, essa frase não passa de só um mito, uma idea antiga e temos de pensar no futuro e sucesso dos fatores globalmente distribuidos.

“A central perde o controlo sobre os seus dados”: com as certas ferramentas comuns e uma continua atualização/comunicação das duas partes, não há perda de dados ou versões de dados.

Com uma metodoogia efectivamente inserida na organização, posso definitivamente vincar que a afirmação “Agile é anti documentação”, não passa de só um mito. Pelo contrário, a metodologias ágil, correctamente implementadas requerem documentação extra.

A minha vivência no estágio possibilitou-me confirmar que estava completamente errada e ensinou-me a distinguir a teoria que é facilmente posta em prática e a teoria que serve só como base para o processo total.



# Capítulo 5

---

## Conclusões

### 5.1 Visão Crítica do Estágio

O estágio consolida-se como o processo que envolve um conjunto de equipas, sujeitos do processo de desenvolvimento. GlobalSoftware foi um dos lugares privilegiados da formação profissional, uma vez que possibilitaram-me uma aproximação inicial e um local para observação direta e desenvolvimento de habilidades no campo profissional.

Com a preocupação de superar a dissociação no estágio, foi preciso pensar em coletivo de trabalho, desenvolvido de forma articulada de construção de conhecimento através de ação concreta que possibilitou compreender as estruturas e os determinantes mais profundos dos projetos. Outro aspecto que se tornou relevante constitui-se no facto de entender o estágio como uma unidade indissociável entre ensino e pesquisa.

Outro tópico diz respeito ao distanciamento existente entre a unidade de Funchal e a central de Londres, pois revela dois tipos de supervisão: a local e a central. Um ponto fraco foi não ter inicialmente um orientador local, apenas a seguir dicas/orientações virtualmente, daí houve uma perda de tempo e desentendimento mútuo. Posso afirmar que tive mais autonomia e própria iniciativa de descoberta. Tive algumas dificuldades em realizar a articulação entre os saberes teóricos e os saberes práticos, em perceber/apreender a realidade na qual atua e mesmo em assumir as funções mas foi uma boa experiência. Na minha opinião, adquiriria mais conhecimentos se tivesse um orientador local, mas os membros da equipa ajudaram-me a superar os problemas com os quais fui confrontada neste percurso.

A minha impressão do estágio foi positiva, a GlobalSoftware tem muitas estratégias em relação à deslocalização de uma forma adequada. Parece que têm trabalhado na capacidade de fazer o desenvolvimento de *software* distribuído globalmente ao longo dos anos, e muitos dos erros típicos de iniciante, como não reconhecer o impacto da distância cultural, não foi encontrado.

A organização é capaz de comunicar de uma forma distribuída com a ajuda das ferramentas de comunicação que tem a capacidade de diminuir a distância em tempo real. Mas um subcapítulo não estudado é o tempo que se perde para receber informações/instruções e a interpretação das mesmas. A distância bloqueia a produtividade máxima invocando sentimentos de desconexão.

Quando as equipas iniciam revisões conjuntas nos locais de trabalho uma da outra, aparecem falhas no progresso.

A equipa beneficia dos progressos tecnológicos, ganhando mais independência ao ponto de sentir-se mais confortável no seu ambiente, transmitindo um efeito positivo na forma de trabalhar, nota-se mais liberdade para tomar decisões.

Uma vantagem notável é a marca conhecida no mercado e a boa reputação. É uma alternativa viável, pois amplia consideravelmente o alcance dos objetivos propostos fornecendo a estrutura necessária para o ingresso na carreira empresarial - o aumento nas oportunidades de sucesso.

O aumento da rentabilidade e redução de custos é um benefício dos recursos financeiros para expandir a sua rede. Com o objetivo de aumentar a sua rentabilidade e também

auxiliar no controle das despesas. Ficam evidentes, também, as vantagens obtidas com as economias de escala e de distribuição aumentando assim a quantidade e reduzindo os seus custos.

A rapidez na expansão conta com a central e a força de trabalho para a instalação e operação das respectivas unidades, torna-se possível a ampliação da rede num ritmo muito mais veloz do que aquele que poderia alcançar caso dependesse apenas de recursos próprios.

Outra vantagem seria a diminuição nas responsabilidades e nos problemas de ordem trabalhista. Sendo cada um juridicamente independente, a responsabilidade em relação aos empregados que trabalham em cada ponto cabe inteiramente, a princípio, àquele que atua neste ponto.

Existe a possibilidade real de juntar as peças na perfeição até aquelas dispersas por locais remotos se for aplicada uma gestão justa. Em prática, as diferenças culturais não afetam o profissionalismo de cada um. Uma boa organização comunica em funcionamento eficiente e equilibrado. Para aumentar a produtividade, a qualidade e a capacidade de enfrentar problemas inesperados, os recursos e as ferramentas têm de partilhar o mesmo nível, assim atingindo o resultado máximo. Se o número de recursos humanos for igual e com mais e melhores ferramentas, a empresa ganha na totalidade. Se deixar uma das divisões na sombra, nos últimos 100 metros notam-se atrasos e falhas que poderiam ser evitados.

Os prazos que o mercado oferece são cada vez mais curtos e com superioridade numa determinada tecnologia.

É menos desafiador gerir esforços discretos do que esforços mais complexos. A maioria das questões nem sempre são comunicadas e é mais complicado avaliar francamente os pontos fracos e fortes das duas partes. Se o conhecimento e a experiência não estiveram alinhados, prolonga-se a curva da aprendizagem.

A falta de comunicação, má interpretação, conversa mais demorosa ou seja a perda de tempo entre outras, haverá sempre limitações, porém elas dependerão do grau de liberdade que fornecem desde o início. Geralmente as restrições do sistema referem-se somente aos aspetos básicos como compra de materiais e equipamentos, criação de novos produtos, abertura de novas unidades e transferência do ponto e do contrato a terceiros.

A distância da outra equipa aproxima-nos da equipa local oferecendo uma cooperação mais forte, uma flexibilidade maior e mais controles sobre as operações.

Considerando a distância, o elevado risco à propriedade intelectual, as diferenças culturais, sistemas jurídicos ou mesmo valores diferentes do local de origem e da central são umas das desvantagens.

O risco de desistência é a desvantagem mais assustadora das enumeradas. Caso ocorra uma diminuição na rentabilidade e lucratividade não prevista no cálculo de retorno de investimento, poderá haver uma eventual desistência.

Embora seja impossível saber mais sobre as culturas e grupos étnicos, é importante aprender o máximo possível. O próprio ato de expressar interesse, melhora a moral e compreensão.

Mas na prática, antes de tudo, foca-se em obter um resultado bem visível do “investimento”, tanto em recursos humanos como também em ferramentas.

Diante de tudo, é possível concluir que a cooperação se constitui em mais uma eficaz colaboração de desenvolvimento e entrega de *software*, possibilitando a ampliação de negócio, por meio de redes de distribuição, aumentando assim o sucesso.

Em suma, unindo o útil ao agradável percebemos que é uma ótima opção de investimento para quem tem um considerável capital disponível e pouca experiência e entendimento na área empresarial, diminuindo-se os riscos de perdas e aumentando-se as possibilidades de sucesso nessa nova carreira.

Partindo do que foi exposto, considero que o estágio caracterizou-se como um lugar coletivo de ensino-aprendizagem e como um sujeito do processo de formação profissional, através de uma prática reflexiva, vivenciada com base no diálogo, na troca, na partilha, na observação, na autonomia, e na sistematização constroem conhecimentos e competências para o exercício da profissão.

## 5.2 Conclusão e Contribuição

A tese de estágio destinou-se a aprofundar empiricamente a problemática dos mitos e realidades como fundamental no desenvolvimento global de *software*. A inserção numa empresa internacional ofereceu a possibilidade de poder fazer as próprias conclusões. E como resultado, apresentámos os desafios e as soluções detalhadas que facilitam o trabalho coletivo em equipas virtuais e o impacto positivo durante a coordenação de projetos.

Como a tecnologia apoia o trabalho global, o *software* torna-se mais onipresente e bastante flexível para as equipas de desenvolvimento virtuais. No entanto foi necessária muita dedicação antes de esse potencial se tornar realidade, novas práticas de trabalho e técnicas de gestão de projetos para elaborar e explorar as diferenças de horário e conquistar os problemas da distância. Dito isso, atualmente tem há série de soluções individuais, tais como ferramentas, práticas e métodos. As realidades que antes do estágio pareciam só mitos continuam a ter a necessidade de serem estudadas.

O estágio foi realizado na empresa GlobalSoftware com uma grande equipa de desenvolvimento de *software*, relativamente estável com um elevado volume de projetos internacionais, embora os desafios decorrentes no dia-a-dia têm um grande peso. Além disso, embora entre a central e a equipa do Funchal não haja diferenças de fuso horário e fluência em Inglês, existem diferenças culturais entre desenvolvedores, *QAs* e *designers*. No entanto, o estágio forneceu muito conhecimento em questões de coordenação relevantes no contexto do desenvolvimento e entrega global de *software*.

As funções assumidas, principalmente a do PM, possibilitaram constatar que é complicado coordenar o trabalho em locais distribuídos mas não impossível. Embora surpreendente, a evidência empírica na literatura é inconclusiva por causa das muitas variáveis de confusão que se correlacionam com a distância geográfica. Além disso, como as tecnologias de colaboração tendo em vista a atenuar a distância continuam a melhorar, os efeitos de dispersão geográfica precisam de uma reavaliação constante. Da minha própria experiência posso constatar que a distância geográfica continua a ser um problema para a

coordenação. Os membros da equipa exigem mecanismos especiais para coordenar o trabalho, dependendo se colaboradores são ou não separados pela distância.

Em suma, a empresa GlobalSoftware de desenvolvimento de *software* global em grande escala beneficia substancialmente em promover o uso de práticas e ferramentas que fortalecem as equipas. Adicionalmente, o meu estudo GDD interno conclui que as diferentes tarefas que têm diferentes tipos de dependências exigem diferentes tipos de coordenação. Portanto, as dependências afetando diferentes grupos em grande escala, as tarefas complexas tais como desenvolvimento de *software* global, precisam ser cuidadosamente compreendidas antes de prescrever remédios de coordenação.

O estudo GDD inicial também contribui para a prática, embora exista muito estudo teórico, a minha realidade, o estar “infiltrado” dentro da GDD e lidar com uma ampla gama de dependências, contribui para essa mesma prática.

### **5.3 Trabalhos Futuros**

O estudo sobre o GDD não estava maduro o suficiente sem o estágio/prática. Nesse contexto, tive a oportunidade de colaborar, superar as distâncias e formar uma conexão com a equipa e os clientes. Outro aspeto que não foi investigado em detalhe e que merece um cuidado mais aprofundado são as questões culturais. No entanto, um dos trabalhos futuros é continuar a analisar o impacto tanto internamente como externamente.

Futuramente uma outra ferramenta de planeamento (Planner) deverá ser usada para gerir as relações entre as tarefas de cada membro da equipa, que serão uma importante adição a documentação externa de um projeto. Outra parte que precisa ser mais trabalhada é a integração entre dos prazos com a duração real de desenvolvimento, já que a capacidade de construção de código é um processo complexo em um projeto.

A minha experiência apresentada neste projeto ainda necessita de uma maior experimentação prática.

# Capítulo 6

---

## Referências

- [1] Herbsleb J., “Global software engineering the future of socio-technical coordination,” in *International conference of software engineering*, Minneapolis, 2007.
- [2] Lanubile F., “Collaboration in distributed software development,” *In Software Engineering*, pp. 14-15, 2009.
- [3] Carmel E. and Agarwal R., “Tactical Approaches for Alleviating Distance in Global Software Development,” pp. 22-29, 2001.
- [4] Herbsleb J.D. and Moitra D., “Global software development,” pp. 16-20, 2001.
- [5] Cataldo M., Bass M. and Herbsleb D., “On coordination mechanisms in global software development,” in *In Global Software Engineering*, 2007.
- [6] Holmstrom H., Fitzgerald B. and Agerfalk J., “Agile practices reduce distance in global software development,” *Information Systems Management*, p. 15, 2006.
- [7] Layman L., Williams L., Damian D. and Bures H., “ Essential communication practices for extreme programming in a global software development team,” *Information and software technology*, p. 781–794.
- [8] Malone W. and Crowston W., “The interdisciplinary study of coordination,” *Computing Surveys*, p. 87–119, 1994.
- [9] Pressman S., *Engenharia de Software 7º Edição*, 2011.
- [10] Ebert C. and De Neve P., “Surviving Global Software Development,” 2001, pp. 62-69.
- [11] Black S. and Jacobs J., “Using web 2.0 to improve software quality,” in

- Proceedings of the 1st Workshop on Web 2.0 for Software Engineering, ACM, 2010.*
- [12] Damian D., Lanubile F. and Oppenheimer H., "Addressing the Challenges of Software Industry Globalization," in *25th International Conference on Software Engineering*, Oregon, 2003.
- [13] Carmel E., *Global Software Teams: Collaborating Across Borders and Time Zones*, 1999.
- [14] Herbsleb J.D. and Moitra D., "Guest Editors' Introduction: Global Software Development," pp. 16-20, 2001.
- [15] Ebert C., Parro C.H., Suttels R. and Kolarczyk H., "Improving validation activities in a global software development," in *Proceedings of the 23rd International Conference on Software Engineering*, Toronto, 2001.
- [16] Highsmith J. and Cockburn A., *Agile software development: the business of innovation*, 2001.
- [17] Chung W. and Alcacer J., "Knowledge sources and foreign investment location in the US," in *Conference of the Academy of International Business*, Monterrey,, 2003.
- [18] Winkler D., Kaltenbach A. and Biffel S., "Evaluating tools that support pair programming in a distributed engineering environment," in *Conference on Evaluation and Assessment in Software Engineering (EASE)*, Keele, 2010.
- [19] Sahay S., "Global software alliances: the challenge of 'standardization'," *Scandinavian Journal of Information Systems*, pp. 3-21, 2003.
- [20] Herbsleb J.D., Paulish D.J. and Bass M., "Global Software Development at Siemens: Experience from Nine Projects.," in *27th International Conference on Software Engineering*, New York, 2005.
- [21] Espinosa J.A. and Carmel E., "The Effect of Time Separation on Coordination Costs in Global Software Teams," in *37th Hawaiian International Conference on System Sciences*, Los Alamitos, 2004.
- [22] Gumm D., "Distribution Dimensions in Software Development Projects: A Taxonomy," pp. 45-51, 2006.
- [23] Boland D., "Transitioning from a Co-Located to a Globally-Distributed Softwar

- Development Team: A Case Study and Analog Devices Inc,” in *Workshop on Global Software Development*, Edinburgh, 2014.
- [24] Damian D. and Zowghi D., “The impact of stakeholders’ geographical distribution on managing requirements in a multi-site organization,” in *IEEE Joint International Conference on Requirements Engineering*, Los Alamitos, 2002.
- [25] Ågerfalk P., “Investigating actability dimensions: a language/action perspective on criteria For information systems evaluation. Interacting with Computers,” 2004, p. 957–988.
- [26] Delone W., Espinosa J.A., Lee G. and Carmel E., “Bridging Global Boundaries for IS Project Success,” in *38th Annual Hawaii International Conference on System Sciences*, Los Alamitos , 2005.
- [27] Gibson C. and Gibbs J., “Unpacking the concept of virtuality: The effects of geographic dispersion, electronic dependence, dynamic structure, and national diversity on team innovation,” *Administrative Science Quarterly*, p. 451–495, 2006.
- [28] Iacovou C.L. and Nakatsu R., “A risk profile of offshore-outsourced development projects,” *Communications of the ACM*, p. 89’94, 2008.
- [29] Kirkman B. and Mathieu J., “The dimensions and antecedents of team virtuality,” *Journal of Management*, p. 700’718, 2005.
- [30] Sakthivel S., “Virtual workgroups in offshore systems development,” *Information and Software Technology*, pp. 305-318, 2005.
- [31] Farshchian B., “Integrating geographically distributed development teams through increased product awareness,” p. 123–141, 2001.
- [32] Sarker S. and Sahay S., “Implications of space and time for distributed work: An interpretive study of US-Norwegian systems development teams,,” *European Journal of Information Systems* , pp. 3-20, 2004.
- [33] Dubé L. and Paré G., “Global virtual teams. Communications of the ACM,” 2001, pp. 71-73.
- [34] Kayworth T. and Leidner D., “The global virtual manager: A prescription for success,” *European Management Journal*, p. 183–194., 2000.

- [35] Gopal A., Mukhopadhyay T. and Krishnan M., "The role of software processes and communication in offshore software development," pp. 193-200, 2002.
- [36] Pauleen D. and Yoong P. , "Relationship building and the use of ICT in boundarycrossing virtual teams: a facilitator's perspective," *Journal of Information Technology*, pp. 2015-220, 2001.
- [37] DeRosa D., Hantula D., Kock N. and D'Arcy J., "Trust and leadership in virtual team work: A media naturalness perspect Warkentin ive," in *Hum. Resour. Manage*, 2004, p. 219-232.
- [38] Krishna S., Sahay S. and Walsham G., "Managing cross-cultural issues in global software outsourcing," *Communications of the ACM*, pp. 62-66, 2004.
- [39] Mortensen M. and Hinds P. , "Conflict and shared identity in geographically distributed teams," *Int. J. Confl. Manage*, p. 221'238, 2001.
- [40] Bell B. and Kozlowski S., "A typology of virtual teams - implications for effective leadership," in *Group & Organization Management*, pp. 14-49.
- [41] Breu K. e Hemingway C.J., "Making organisations virtual: the hidden cost of distributed teams," *Journal of Information Technology*, pp. 191-202, 2004.
- [42] Battin R., Crocker R., Subramanian K. and Kreidler K., "Leveraging Resources in Global Software Development," 2001, pp. 70-77.
- [43] Evaristo J.R., Scudder R. and Desouza K.C., "A dimensional analysis of geographically distributed project teams: a case study," *Journal of Engineering and Technology Management*, pp. 175-189, 2004.
- [44] Välimäki A., Koskimies K. and Kääriäinen J., "Global software development patterns for project management," *Global IT Outsourcing Study. DiamondCluster International, Inc.*, pp. 137-148, 2009.
- [45] Hossain E., Ali B. and June V., "How can agile practices minimize global software development co-ordination risks? In: Software Process Improvement, Communications in Computer and Information Science," p. 81-92, 2009.
- [46] Poston R. and Sexton M. P., "Evaluating and selecting testing tools," 1992, p. 33-42.
- [47] Hofstede G., "Culture's Consequences: International Differences in Work-

- Related Values,” pp. 6, 60, 61, 87, 133, 2001.
- [48] Robinson M. and Kalakota R., “Offshore Outsourcing: Business Models, ROI and Best Practices,” 2004.
- [49] Nurick A. and Thamhain H. J. , “Developing Multinational Project Teams. In: Global Project Management,” in *Planning, Organizing and Controlling International Projects*, New York, 2006.
- [50] PMI, A Guide to the Project Management Body of Knowledge (PMBOK®), Project Management Institute, 5th Ed. .
- [51] Kotlarsky J. and Kumar K., “Coordination and Collaboration for Globally Distributed Teams: The Case of Component-Based / Object-Oriented Software Development,” 2002.
- [52] Lamersdorf A. and Münch J., “A tool for supporting task distribution in global software development projects,” in *International workshop on tool support development and management in distributed software projects, Proceedings of the fourth international conference on global software engineering*, 2009.
- [53] Abrahamsson P., Salo O., Ronkainen J. and Warst J., Agile software development, 2002.
- [54] “Manifesto for Agile Software Development,” [Online]. Available: <http://www.agilemanifesto.org>.
- [55] Damian D., Williams L., Layman L. and Bures H., “Essential communication practices for extreme programming in a global software development team,” in *Information & Software Technology* , 2006, p. 781–794.
- [56] Xiaohu Y., Bin X., Zhijun H. and Maddineni S., “Extreme programming in global software development,” in *Electrical and Computer Engineering*, 2004, p. 1845–1848.
- [57] Ramesh B., Cao L., Mohan K. and Xu P., “Can distributed software development be agile?” SPECIAL ISSUE: Flexible and distributed software processes: old petunias in new bowls,” 2006, pp. 41-46.
- [58] Paasivaara M., Durasiewicz S. and Lassenius C., “Distributed agile development: Using Scrum in a large project. Software Process Improvement and Practice,” p. 527–544 , 2008.

- [59] Cockburn A. and Highsmith J., "Agile Software Development: The Business of Innovation," pp. 120-122, 2011.
- [60] Charette R., "Fair Fight? Agile Versus Heavy Methodologies," *Cutter Consortium E-Project Management Advisory Service*, 2001.
- [61] Sutherland J. and Schwaber K., "The Scrum Papers: Nuts, Bolts and Origin of an Agile Process," 2009.
- [62] Carmel E. and Agarwal R., "Tactical approaches for alleviating distance in global software development," pp. 14-15,54, 2001.
- [63] G. Hofstede, "Culture's Consequences: International Differences in Work-Related Values," pp. 6, 60, 61, 87, 133, 2001.
- [64] Sakthivel S., "Virtual workgroups in offshore systems development," *Information and Software Technology*, pp. 305-318.
- [65] Paasivaara M. and Lassenius C., "Using iterative and incremental processes in global software development," in *International Workshop on Global Software Development', ICSE Workshop*, London, 2004.