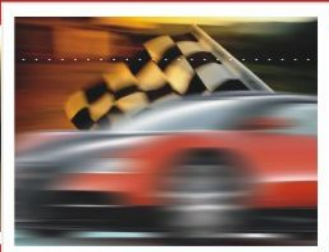




Scrum Certificado (MPS.Br F)

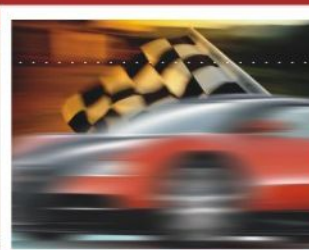
A experiência da Powerlogic



Paulo Alvim
(alvim@powerlogic.com.br)



Sobre a Powerlogic



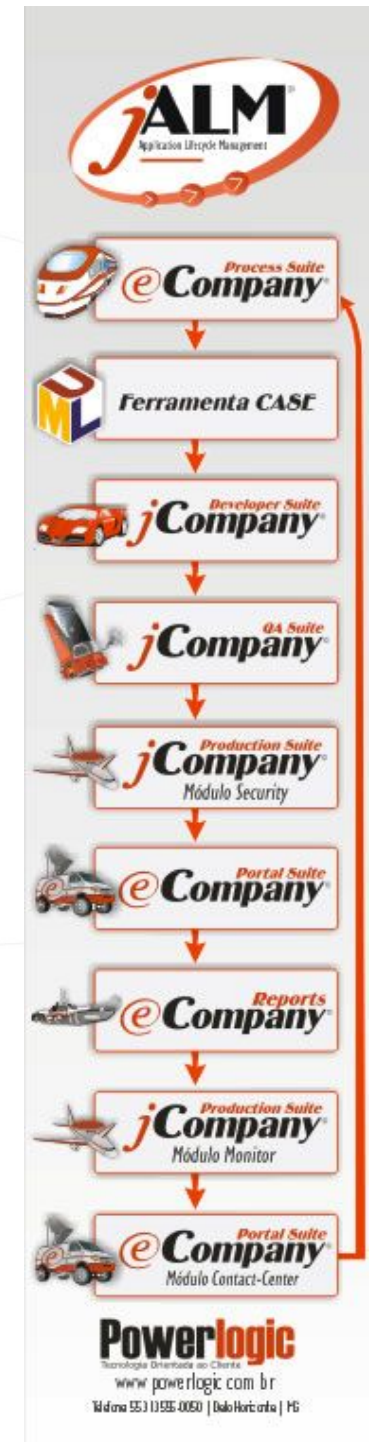
Paulo Alvim
(alvim@powerlogic.com.br)

Quadro Diretor

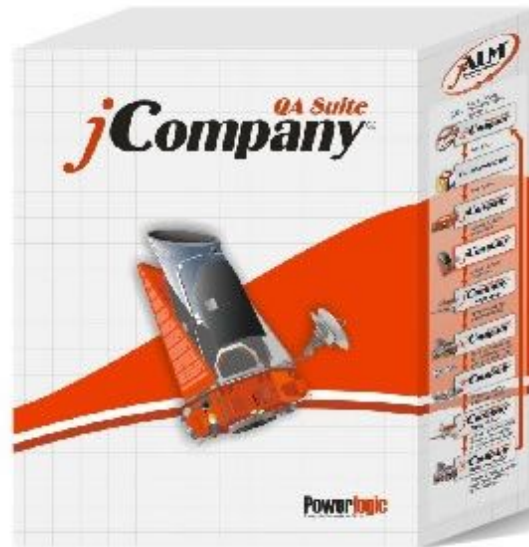
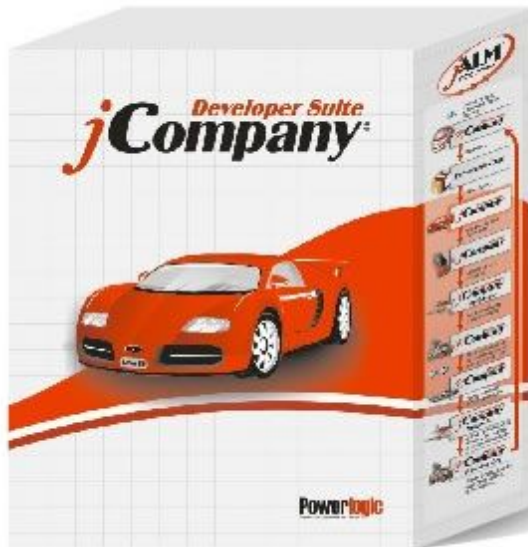
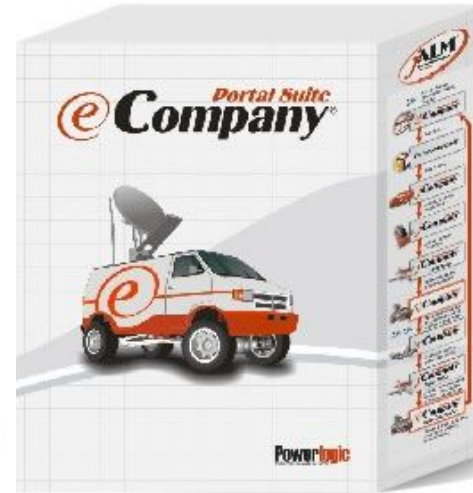
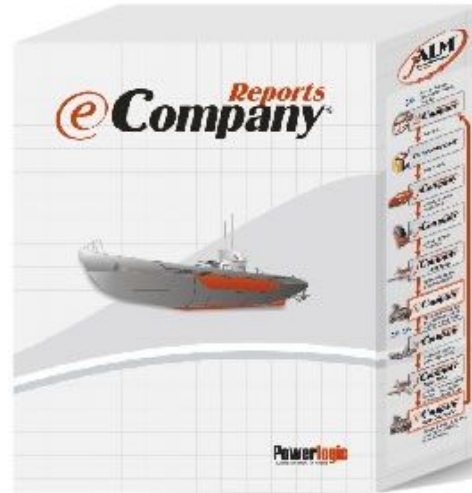
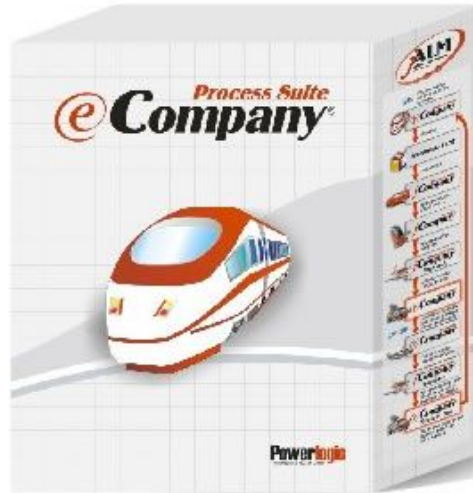
- ❖ 1987: Graduação na UFMG; *Expertise* em Engenharia da Informação.
- ❖ 1987-1991: Ferramenta CASE ET-SADS; Desenv. Mainframe.
- ❖ 1992-1994: Orientação a Objetos/Eventos (Unisys Ally for Unix, PowerBuilder)

Histórico Powerlogic

- ❖ 1994 (Dezembro): Fundação, com foco pragmático em downsizing
- ❖ 1995-1998: Desenvolvimento C/S com RUP e variações
- ❖ 1999-2000: Desenvolvimento Java; Lançamento do eCompany Portal Suite e eCompany Project Suite.
- ❖ 2001-2003: Desenvolvimento J2EE Open-Source; **Primeiras incursões em Agile** (Palestra “Scrum” XP/SP em 2002)
- ❖ 2003: Lançamento do **jCompany Developer Suite 1.0**
- ❖ 2004: Aporte Capital BNDES (**Sociedade Anônima**)
- ❖ 2004-2007: Evolução e Integração de produtos no **Powerlogic jALM 5.1**
- ❖ 2007 (Junho): **Certificação MPS.Br Nivel F** (CMMI 2)
- ❖ 2008: Lançamento do jCompany Free e Professional 5.1; Livro “Tirando o Máximo do Java EE 5”
- ❖ 2008 (Março): Início da Certificação MPS.Br Nivel C (CMMI 3)
- ❖ Faturamento 2008 projetado em 12M; 160 funcionários, 30 profissionais na área de em produtos



Powerlogic jALM



Cliente Feedback

Suporte: erros (bugs), dúvidas, sugestões de melhorias, novas demandas...

Processo, Produtos, Componentes, Projetos, Requisitos

Gestor

Suporte
Portal Suite @Company
 Contact-center



Planejamento
Process Suite @Company



Aplicação monitorada, contabilização em uso, auditoria, estatísticas. Disponibilidade e ROI!

Monitoria
Production Suite jCompany
 Monitor



Elaboração
ENTERPRISE ARCHITECT



Java, modelos, ...

Relatórios Por Demanda

BI
Reports Suite @Company



Fase Retorno

Fase Investimento

Full Application LifeCycle including ROI

Construção
Developer Suite jCompany



Aplicação executável, códigos fontes (WAR, EAR) .

Qualidade
QA Suite jCompany



Aplicação contextualizada para usuários, apoiada por utilitários de colaboração.

Aplicação averiguada e segura (produção).

Aplicação e práticas averiguadas ...

Colaboração
Portal Suite @Company

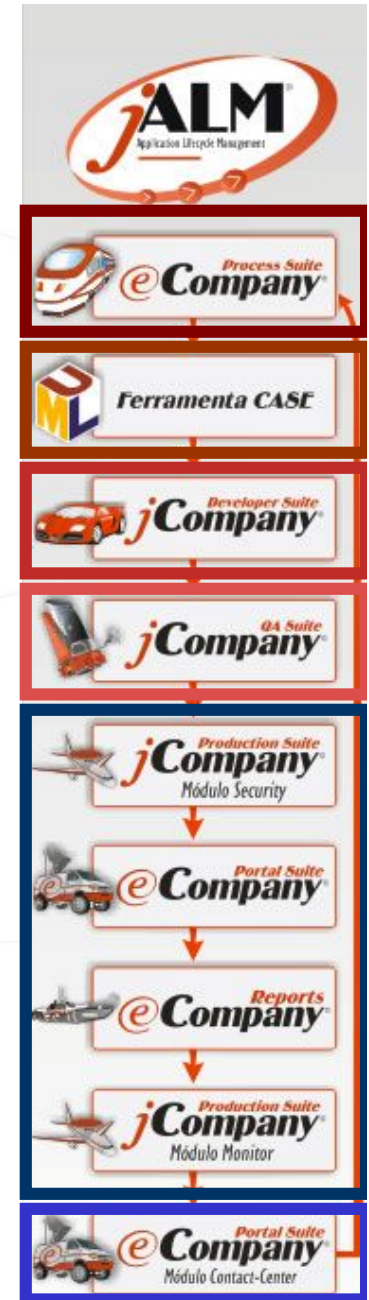
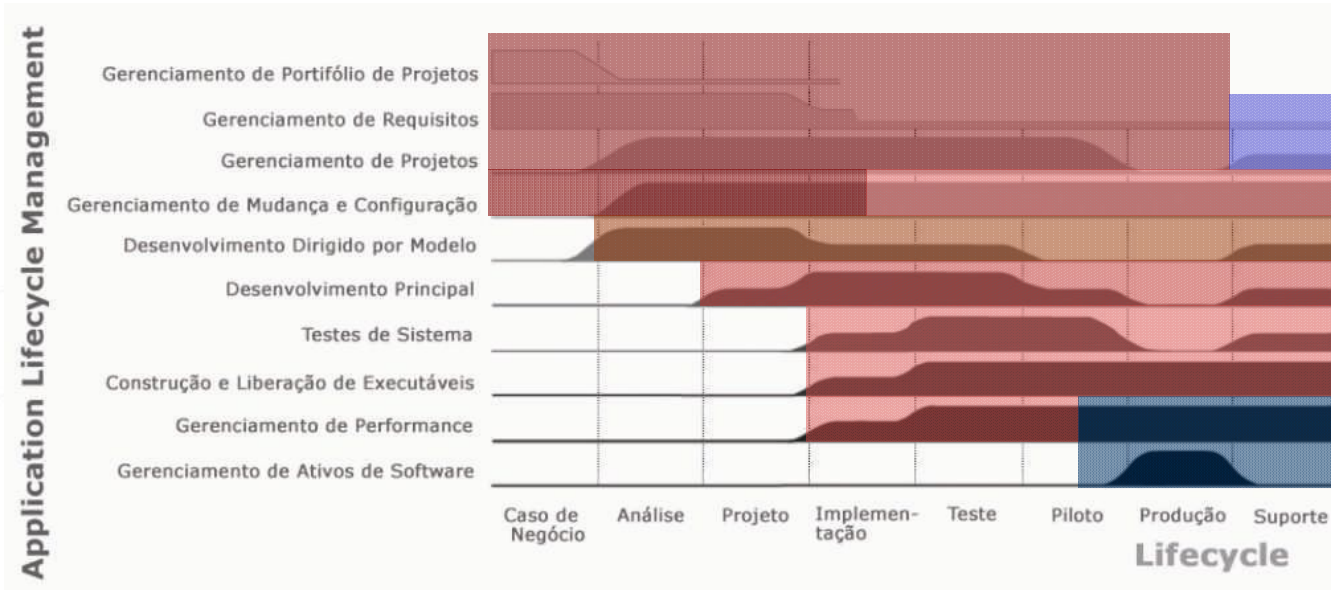


Pré-produção
Production Suite jCompany
 Security



Cliente

Powerlogic jALM



Powerlogic
Tecnologia Orientada ao Cliente

Clientes



VALLOUREC & MANNESMANN TUBES



Powerlogic

Tecnologia Orientada ao Cliente

Clientes

IAG Saúde
Instituto de Acreditação e Gestão em Saúde



Celuloso Nipo-Brasileira S.A.

ARAUJO
DROGARIA DRUGSTORE



fumsoft
Sociedade Mineira de Software

magnesita
service



Magnesita S.A.

CAROL



Powerlogic

Tecnologia Orientada ao Cliente

Clientes



Powerlogic

Tecnologia Orientada ao Cliente

Clientes



TRT 12ª Região
Santa Catarina



J.MACÊDO



FARMÁCIAS
Pague
Menos
DRUGSTORE

TRIBUNAL REGIONAL DO TRABALHO DA 4ª REGIÃO
RIO GRANDE DO SUL



Solution Providers

Belém
PD Case

Recife
Vinic

Vitória
Acrópolis

Porto Alegre
Ibrowse

São Luís
PD Case

Maceió
RCA
Vinic

Curitiba
Elosoft

Goiânia
Avanti

Teresina
RCA

Fortaleza
RCA

Rio de Janeiro
Avanti
Ibrowse

Brasília
Avansys
7COMm
Plancorp
Avanti

Natal
RCA
Vinic

Aracaju
Avansys
Vinic

Campinas
Ibrowse

João Pessoa
RCA
Vinic

Salvador
Avansys

São Paulo
7COMm
Prime
Plancorp

Campo Grande
DSF

Cuiabá
DSF

Belo Horizonte
PD Case
Save

Florianópolis
Elosoft

Blumenau
Elosoft

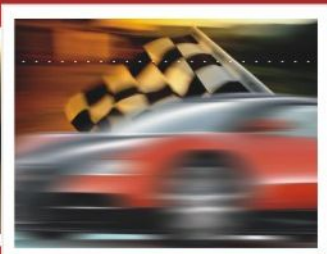
Cobertura



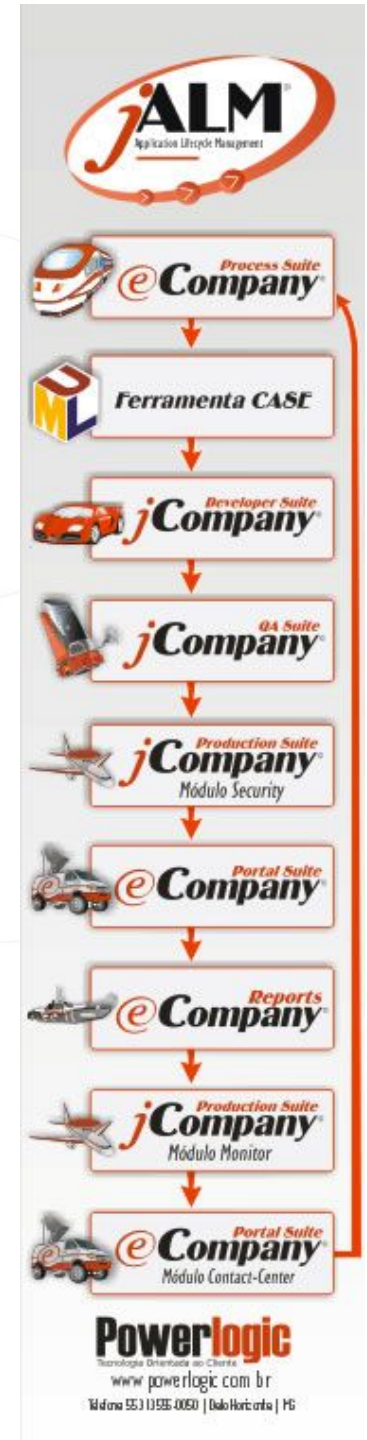
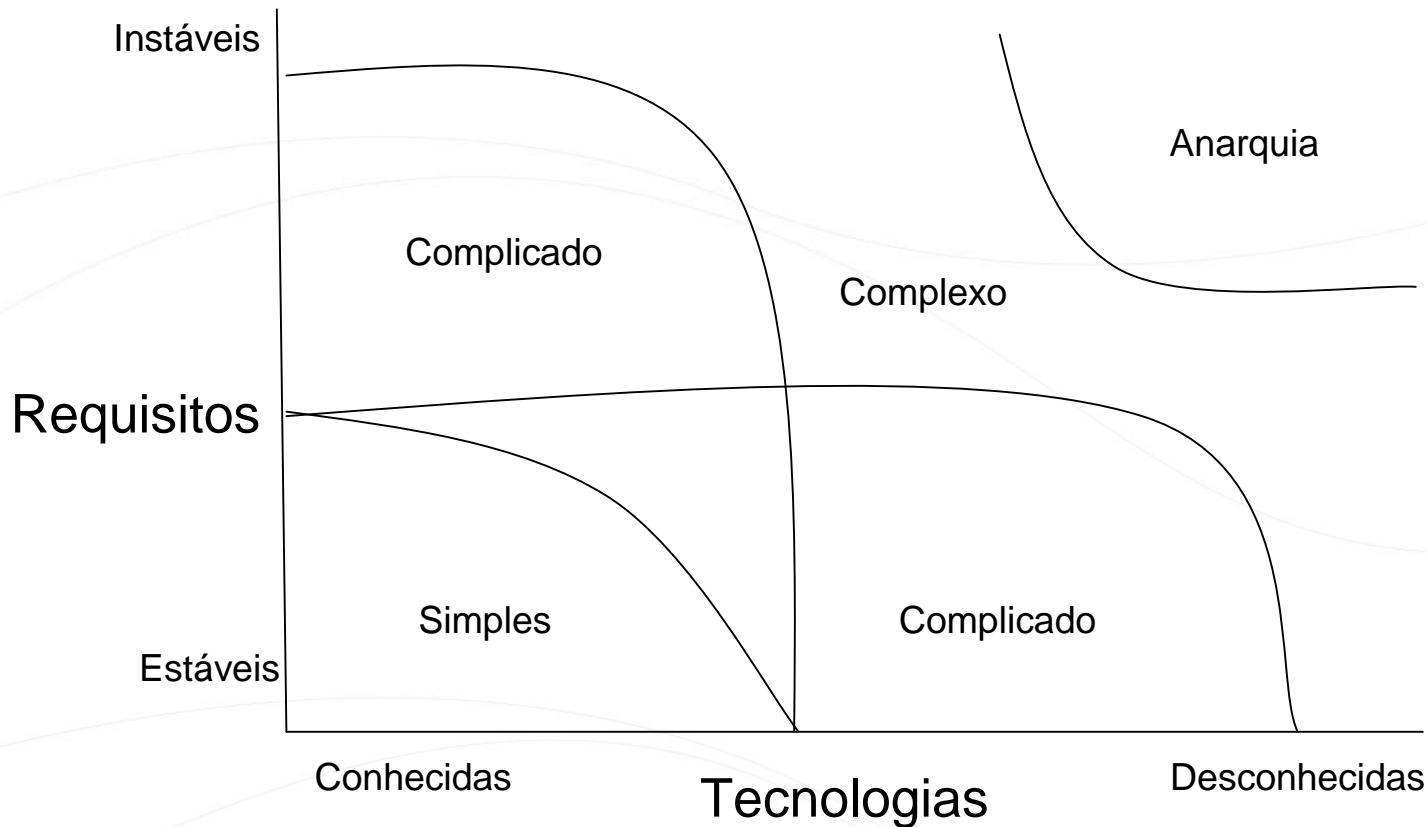


Lições Aprendidas em Processos de Desenvolvimento de Software (PDS)

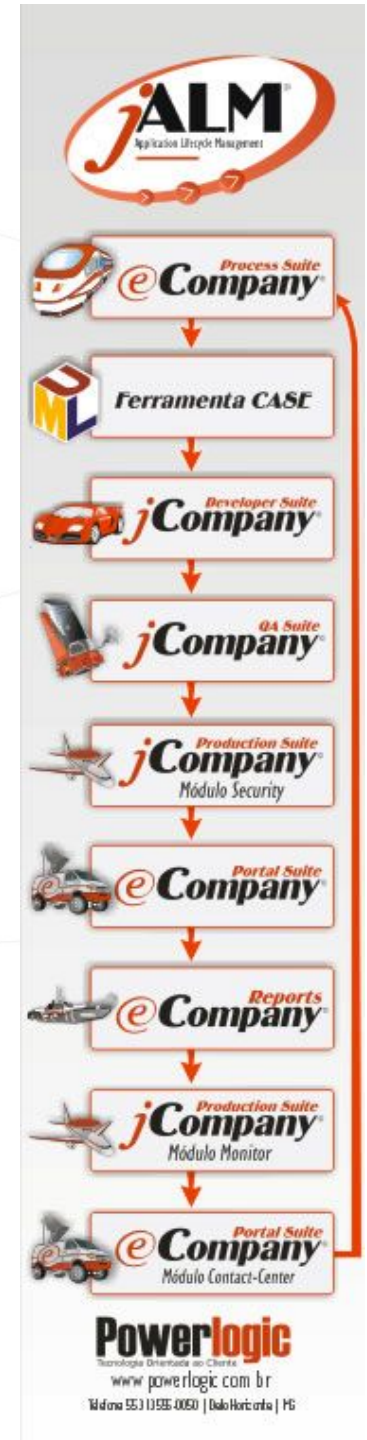
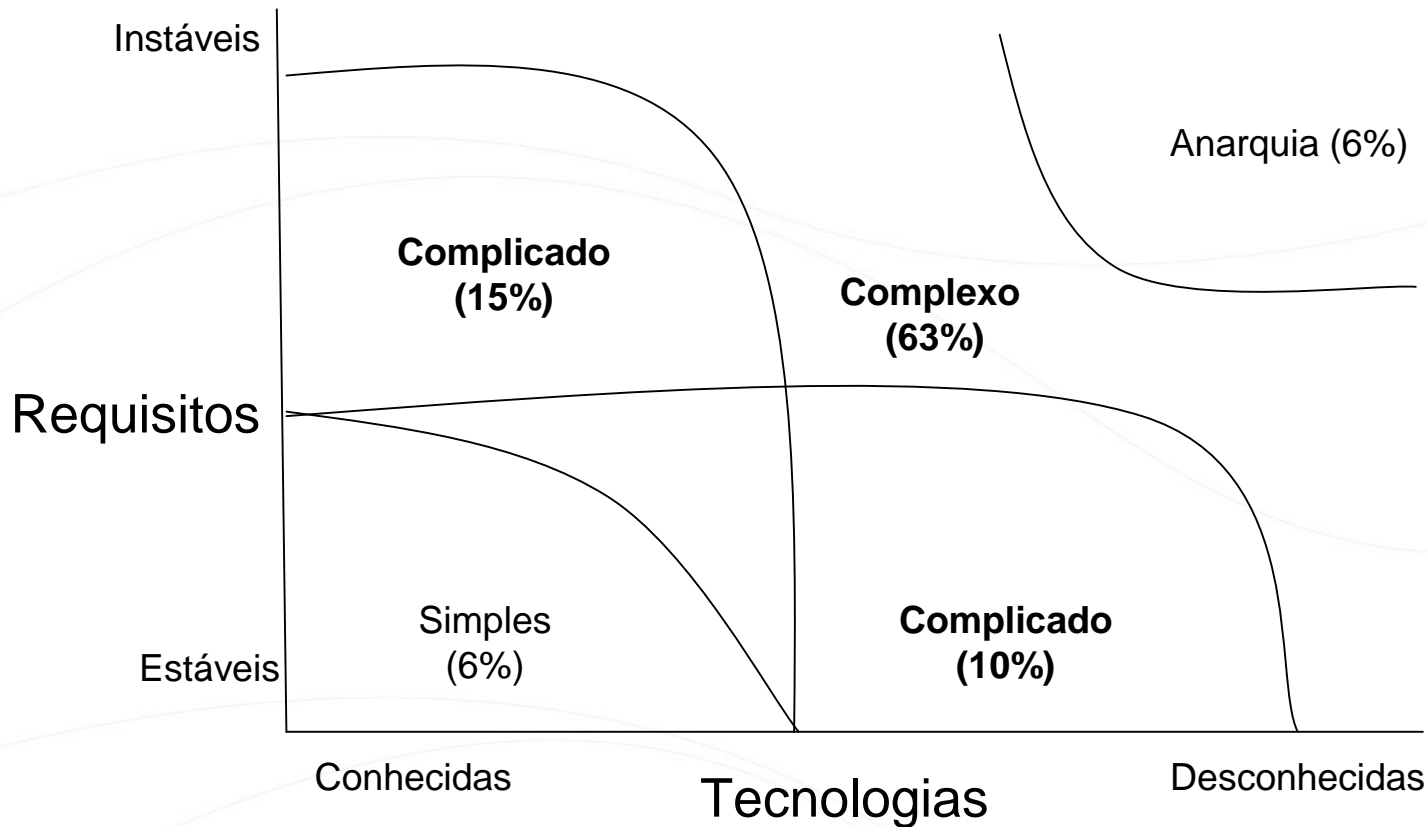
A experiência da Powerlogic: Décadas de 80 e 90



“Processos em Cascata (Preditivos) não são adequados para o desenvolvimento de softwares complexos...”



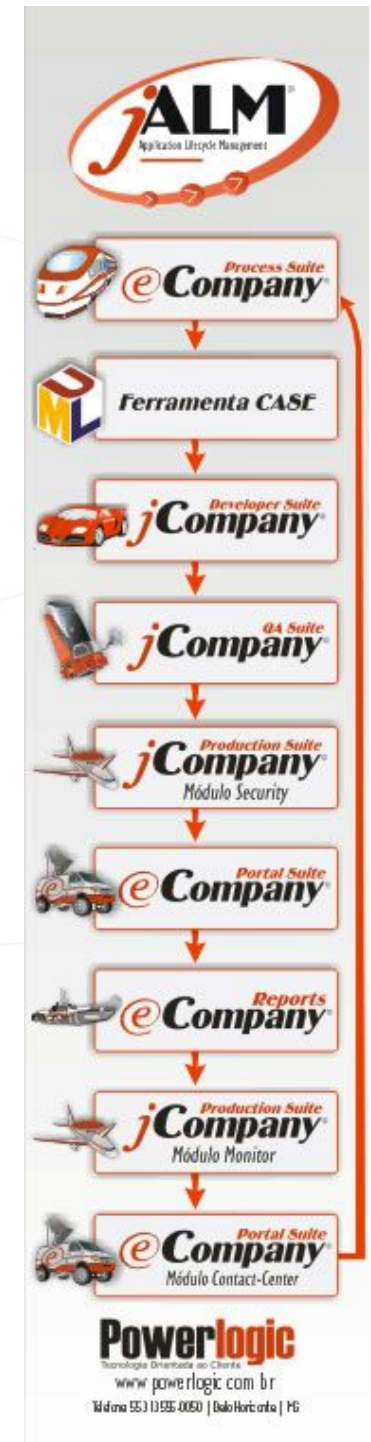
“... e a maior parte dos projetos corporativos na atualidade caem nesta categoria”



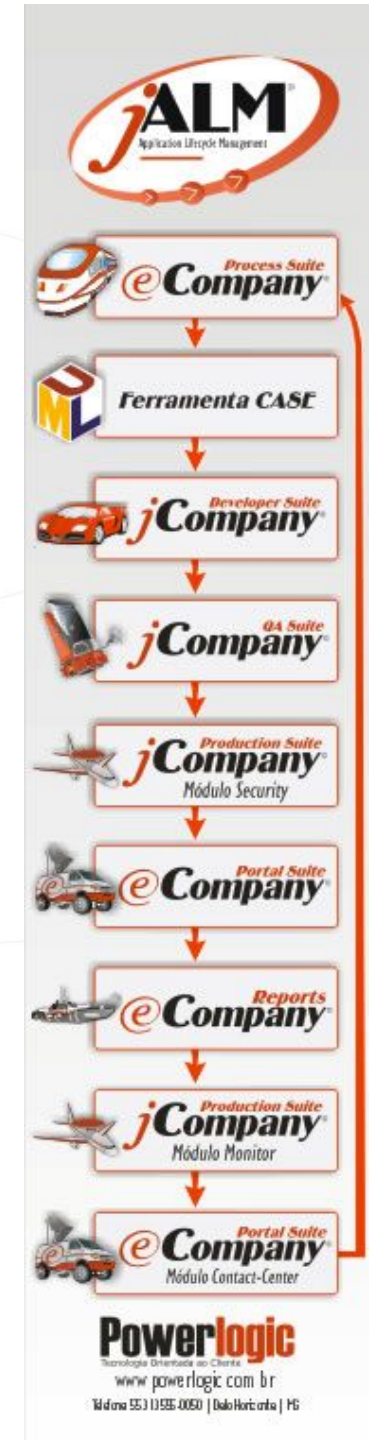
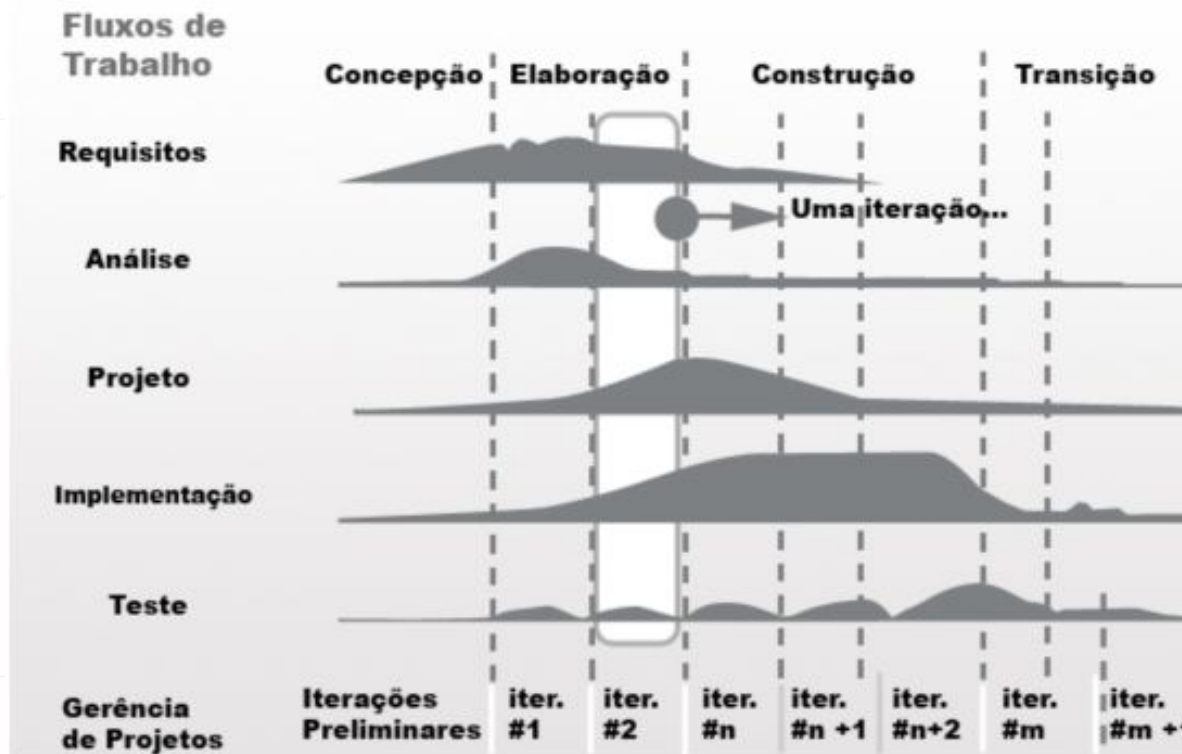
“Projetos complexos são melhor gerenciados por **Processos Empíricos e Iterativos.**”

Estados de um Processo (Teoria do Caos):

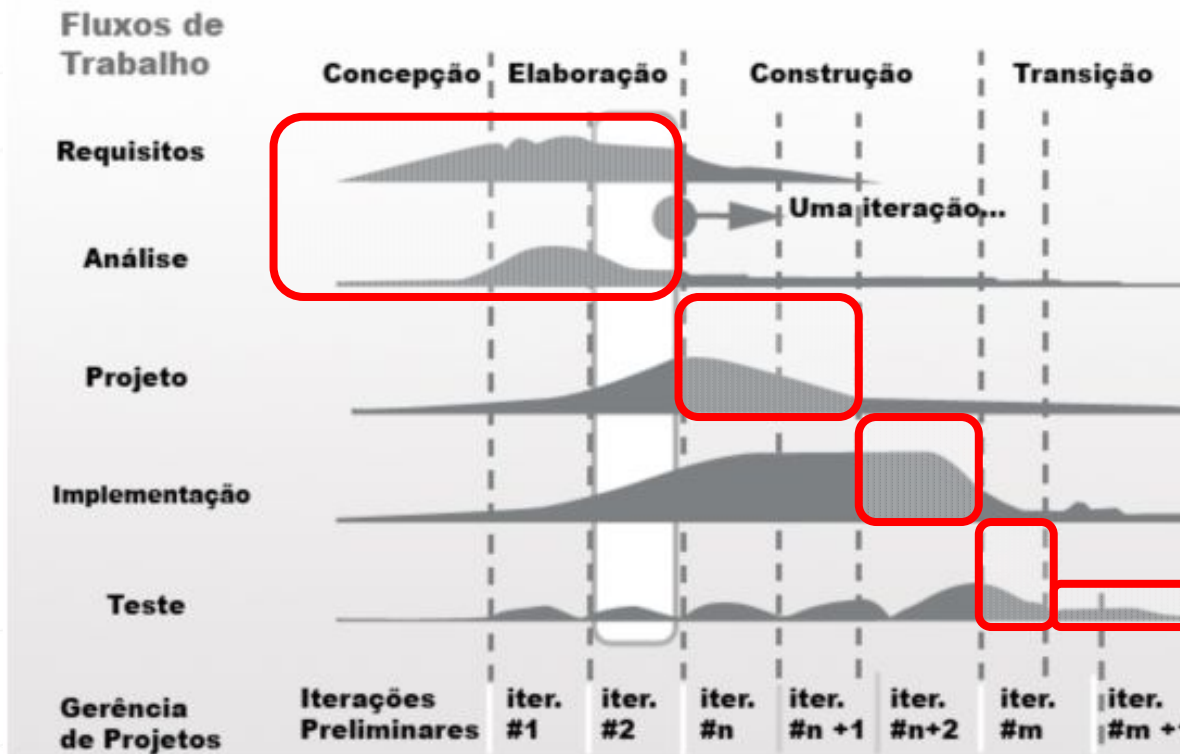
- ❑ **Ideal:** Entradas, Saídas e Variáveis de Processo Estáveis
→ Desenvolvimento de Software nunca está neste estado!
- ❑ **Limítrofe:** Processos Controláveis Estatisticamente de Forma Razoável. Variâncias em número pequeno, previsíveis e gerenciáveis.
→ Desenvolvimento de Software está eventualmente neste estado!
- ❑ **Beira do Caos!** Ruídos severos. Tolerâncias fora das aceitáveis. Previsibilidade e planos quase ineficazes. **Observação contínua pode liberar produtos convergentes!**
→ Desenvolvimento de Software está usualmente neste estado!
- ❑ **Caos!** Processos sem controle, que não resultam em produtos convergentes (em conformidade com o esperado).
→ Isto pode soar familiar para muitos desenvolvedores de software!



“O RUP (*Rational Unified Process*) foi o primeiro Processo Iterativo a ser popularizar, concebido como contraposição à ‘Paralisa da Análise da Engenharia da Informação’...”



“... no entanto, sua extensão e complexidade usualmente distraem a atenção sobre **seu caráter Iterativo!** A grande maioria das implementações de UP terminam como ‘bisnetos UML dos Processos em Cascata’.”



Os “montinhos” que representam a iteratividade no diagrama clássico do UP, após 15 anos, ainda são mal interpretados

Stack de produtos Powerlogic:

- jALM (Application Lifecycle Management)
- @Company Process Suite
- Ferramenta CASE
- jCompany Developer Suite
- jCompany QA Suite
- jCompany Production Suite Módulo Security
- @Company Portal Suite
- @Company Reports
- jCompany Production Suite Módulo Monitor
- @Company Portal Suite Módulo Contact-Center

Powerlogic
Tecnologia Orientada ao Cliente
www.powerlogic.com.br
Tel: (11) 5531-1535-0050 | Belo Horizonte | MG

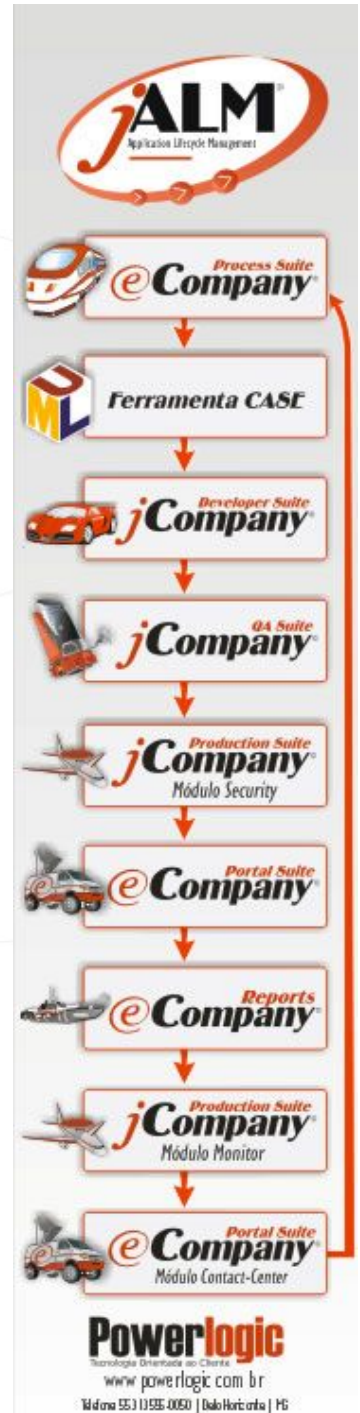
“Contratos ‘Escopo Fechado’ x Necessidades Técnicas: Um fator de contribuição para este retrocesso é o **modelo de Fábrica de Software**, que pressupõe a separação física de especialidades (Ex.: Analista de Negócio x Desenvolvedor), dificultando ou impossibilitando uma abordagem verdadeiramente Iterativa, que permita, por exemplo, construção em estágios iniciais do processo”

Contratante | Fábrica

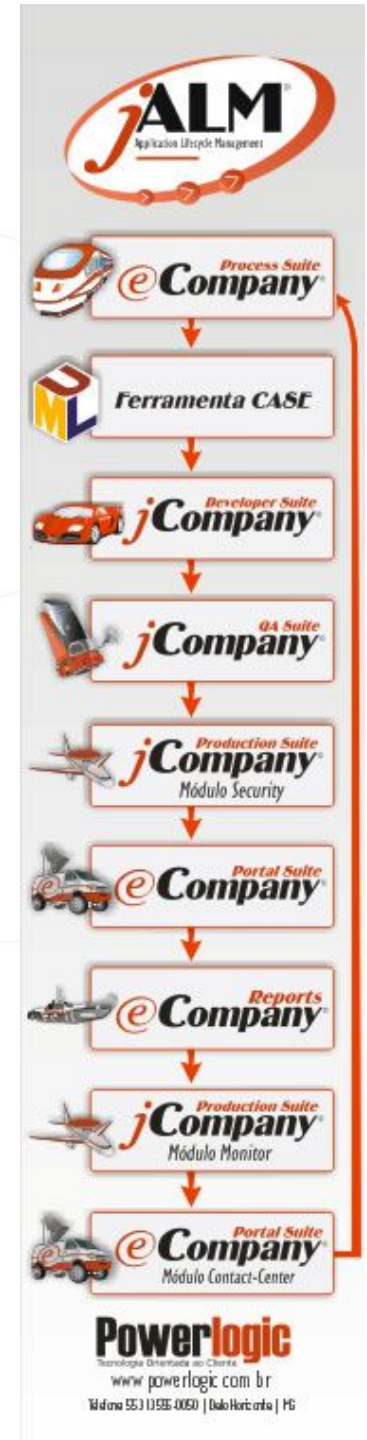
“Eu conheço o negócio, escrevo especificações e contratos, mas não conheço tecnologias e nem como construir”

“Eu conheço tecnologias e sei como construir, mas não conheço o negócio. Leio a especificação e executo o que me pedem.”

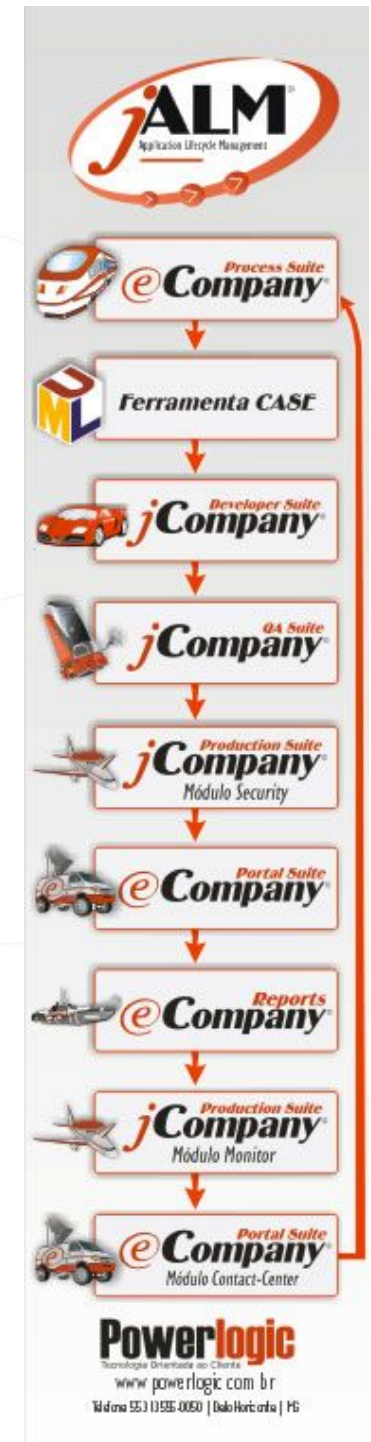
Conhecimentos complementares
distanciados por
ênfase em ‘Contrato de Escopo Fechado’



“Software é 50% introspecção e 50% comunicação.
Comunicação por escrito é boa para criação de memória, mas o canal menos eficaz para obtenção de entendimento, quando comparada a meios interativos”



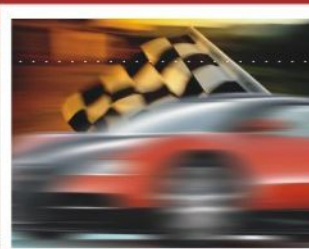
“Documentação realmente útil em desenvolvimento de software é obtida **‘por consequência de atividades produtivas’**, e não **‘como um esforço à parte’**. Atividades meramente documentacionais tendem ao desperdício (documentação *‘write-only’*)”





Lições Aprendidas em Sete Anos de Processos Ágeis

A experiência da Powerlogic: 2001-2008



Paulo Alvim
(alvim@powerlogic.com.br)

Por que Ágil?

“O **Manifesto da Agilidade, publicado em 2001**, resume de forma sensata (e até brilhante!) a experiência de seus 17 autores em projetos reais de desenvolvimento de software, durante as décadas de 80 e 90 - sendo consistente com a experiência da grande maioria de empresas como a Powerlogic, que atuam com **‘metodologia aplicada, focada em resultados’**”

We are uncovering better ways of developing software by doing it and helping others do it.
Through this work we have come to value:

- Individuals and interactions over processes and tools
- Working software over comprehensive documentation
- Customer collaboration over contract negotiation
- Responding to change over following a plan

That is, while there is value in the items on the right, we value the items on the left more.

| | | |
|-------------------|----------------|------------------|
| Kent Beck | James Grenning | Robert C. Martin |
| Mike Beedle | Jim Highsmith | Steve Mellor |
| Arie van Bennekum | Andrew Hunt | Ken Schwaber |
| Alistair Cockburn | Ron Jeffries | Jeff Sutherland |
| Ward Cunningham | Jon Kern | Dave Thomas |
| Martin Fowler | Brian Marick | |



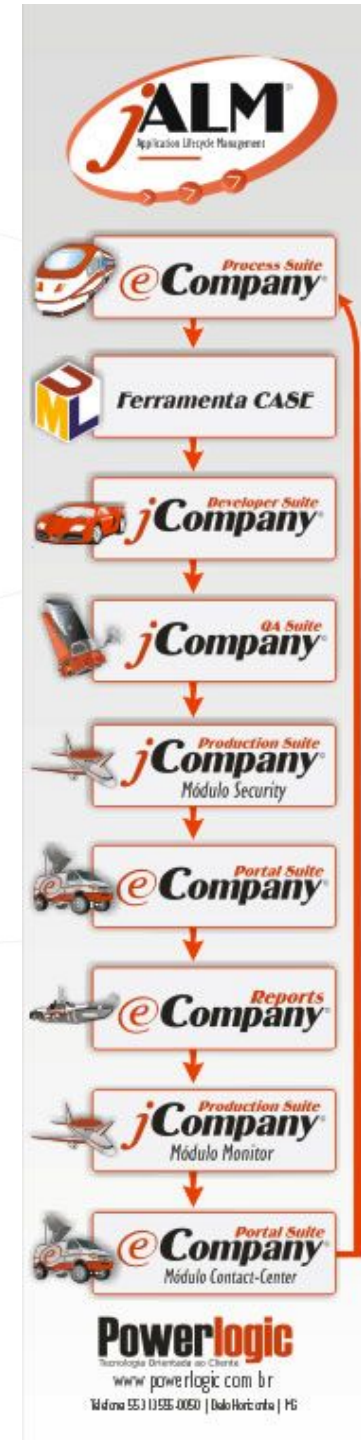
“O caráter sensato “de ênfase” (e não ruptura) do Manifesto da Agilidade tem sido ignorado em algumas posturas radicais que, precipitadas, chegam a confundir **agilidade** com mera **informalidade**.”

We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:

Individuals and interactions over processes and tools
 Working software over comprehensive documentation
 Customer collaboration over contract negotiation
 Responding to change over following a plan

That is, while there is value in the items on the right, we value the items on the left more.

| | | |
|-------------------|----------------|------------------|
| Kent Beck | James Grenning | Robert C. Martin |
| Mike Beedle | Jim Highsmith | Steve Mellor |
| Arie van Bennekum | Andrew Hunt | Ken Schwaber |
| Alistair Cockburn | Ron Jeffries | Jeff Sutherland |
| Ward Cunningham | Jon Kern | Dave Thomas |
| Martin Fowler | Brian Marick | |

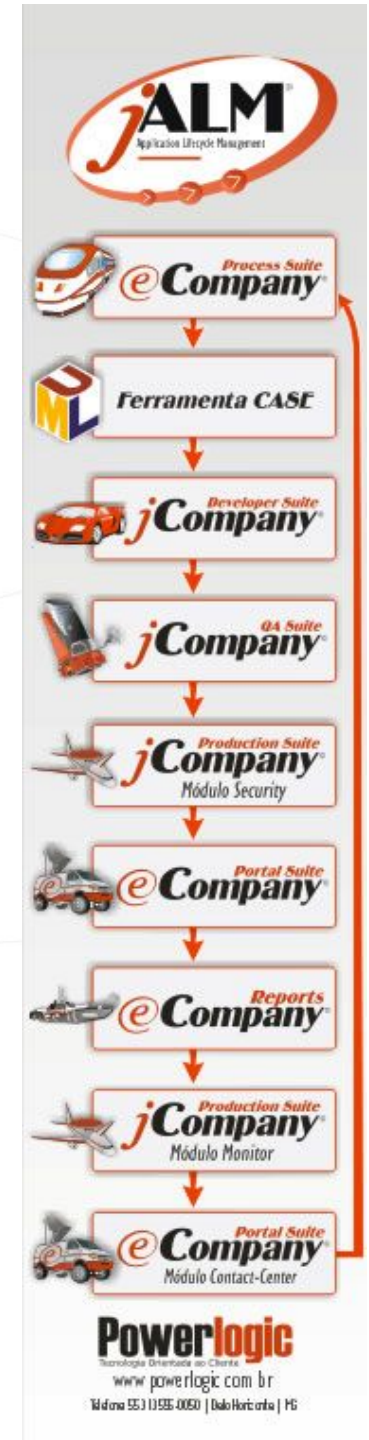


“Apesar dos valiosos estudos da escola ágil sobre a natureza do Processo de Desenvolvimento de Software, **em um nível inédito de profundidade e de coerência** entre diversos especialistas pragmáticos, a percepção de seu valor tem sido eclipsada por posturas simplórias...”

Some Managers perception of plan driven and agile methods



Plan Driven



“Apesar dos valiosos estudos da escola ágil sobre a natureza do Processo de Desenvolvimento de Software, **em um nível inédito de profundidade e de coerência** entre diversos especialistas pragmáticos, a percepção de seu valor tem sido eclipsada por posturas simplórias...”

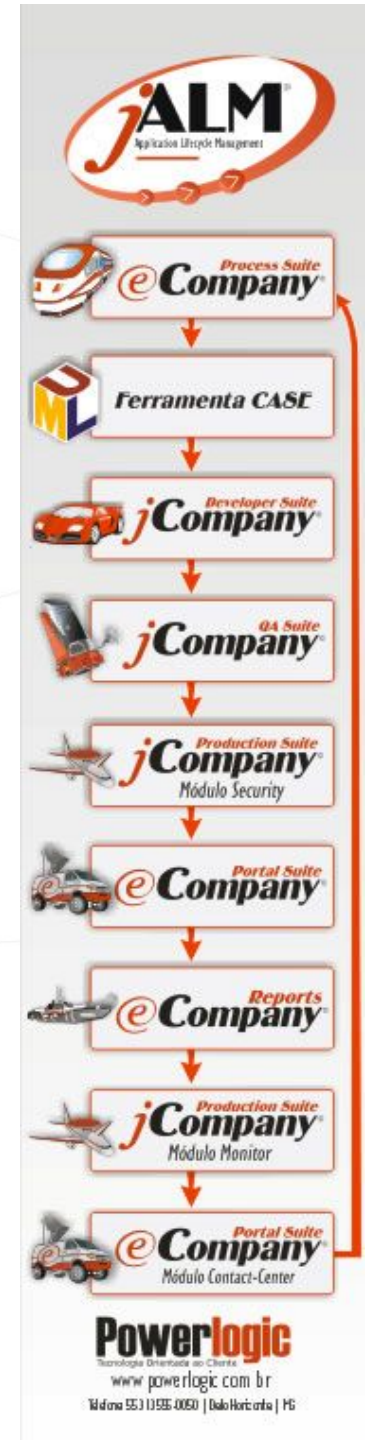
Some Managers perception of plan driven and agile methods



Plan Driven



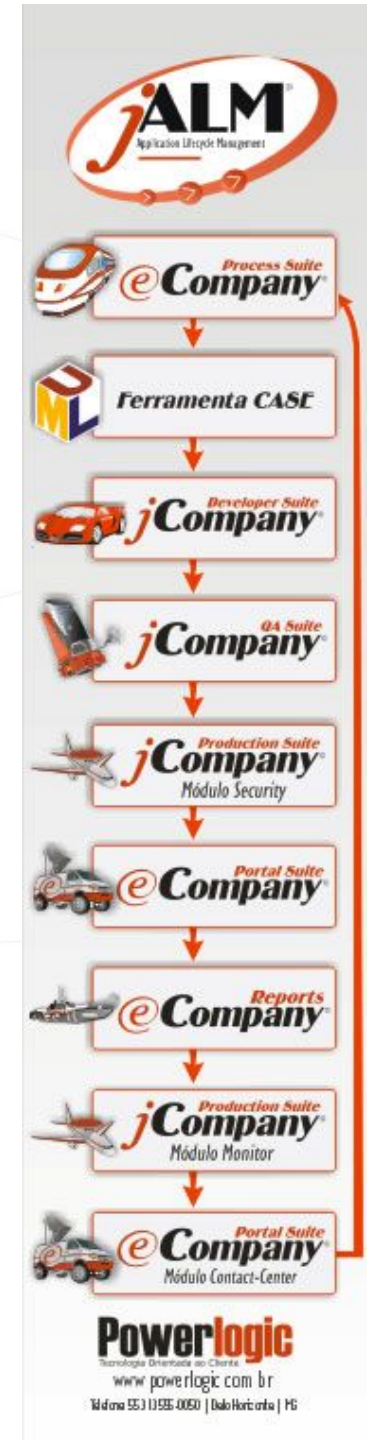
Agile



“Em entrevista pela ocasião do aniversário de seis anos do Manifesto da Agilidade, **Scott Ambler reafirmou o valor do manifesto**, amparado por estatísticas de crescimento, adoção e sucesso. Porém, já apontou duas pequenas revisões como “experiências adquiridas”, neste período:

Indivíduos e Interações sobre Processos e Ferramentas.

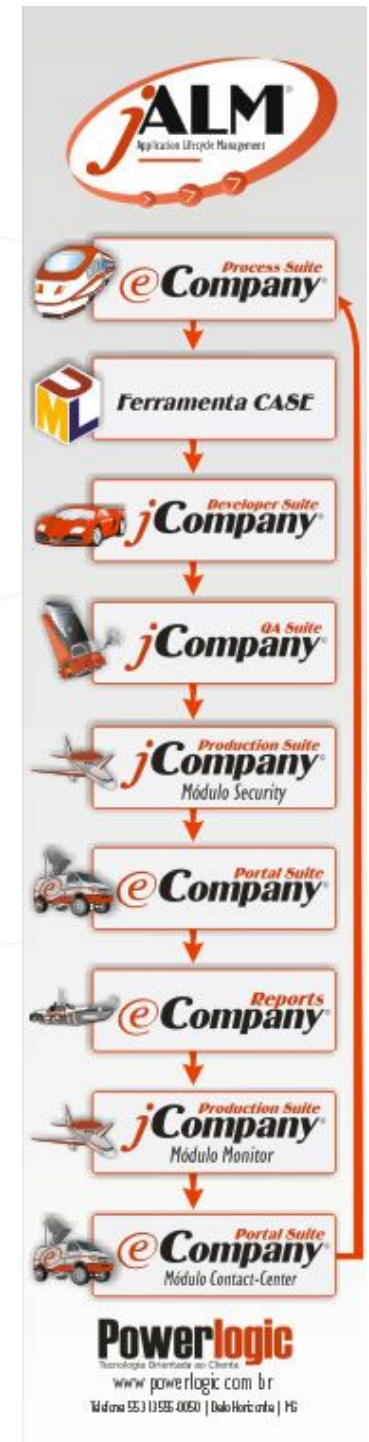
- “Indivíduos e Interações” continuam mais valiosos do que “Processos e Ferramentas”, **mas muitas ferramentas se mostraram fundamentais para práticas de Engenharia de Software Ágeis**, tais como Refactoring, Integração Contínua e TDD – sem a evolução e popularização de IDEs e de funcionalidades nestas áreas, seria difícil postular seu valor.
- Métodos Ágeis exigem um quadro técnico (Indivíduos) de perfil diferenciado. **Não espere “auto-organização” e soluções técnicas com alta conformidade de equipes frágeis, de perfil mediano.** A ênfase em “indivíduos e interações” exige exatamente isso: ênfase na boa formação de “times” e na preservação de talentos.



“Processos Ágeis são mais leves (preconizam menos práticas), porém exigem alto índice de disciplina. Suas motivações são mais difíceis de se explicar (teoria das restrições, teoria do caos, comunicação e colaboração, ...) e, por consequência, de se compreender.”

“São exemplos de práticas que levaram um bom tempo para serem incorporadas na Powerlogic:

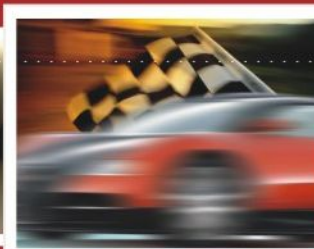
- Antes do MPS.Br, muitas práticas não eram (honestamente) valorizadas pelas equipes, tais como a integração contínua, os 30 dias fixos de um ciclo Scrum (Sprint) ou o próprio rigor com o Daily Scrum.
- O uso apropriado do “Agile Radiator” (quadros brancos e post-its) também sofreu diversas revisões e aprimoramentos, ao longo dos anos, após uma melhor compreensão de seu valor pelas equipes.





A Certificação MPS.Br na Powerlogic

2006-2007

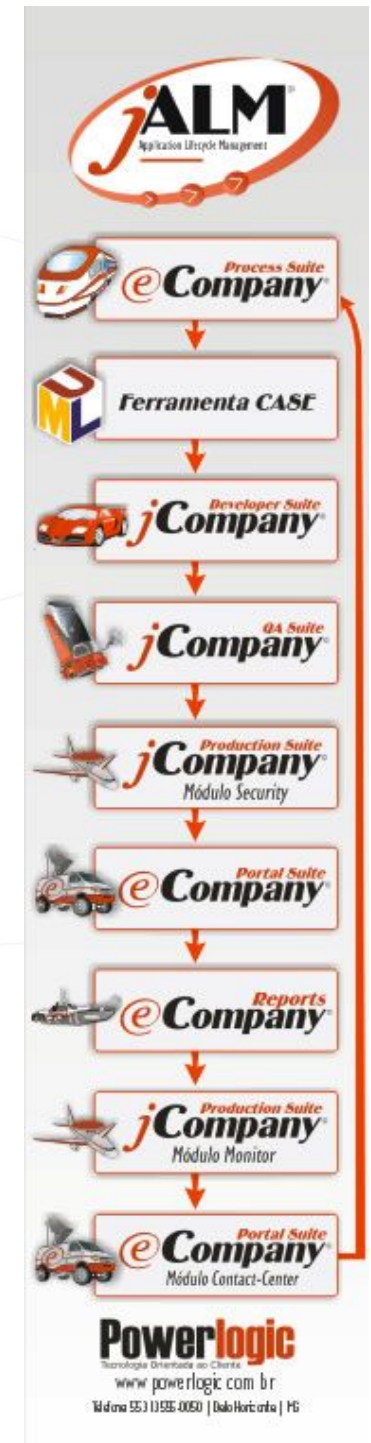


Paulo Alvim
(alvim@powerlogic.com.br)

“Certificações como CMMI (e MPS.Br) são direitos válidos que o mercado possui de exigir dos fornecedores uma demonstração reconhecível de seu esforço em estabelecer, gerenciar e aprimorar seus PDS.”

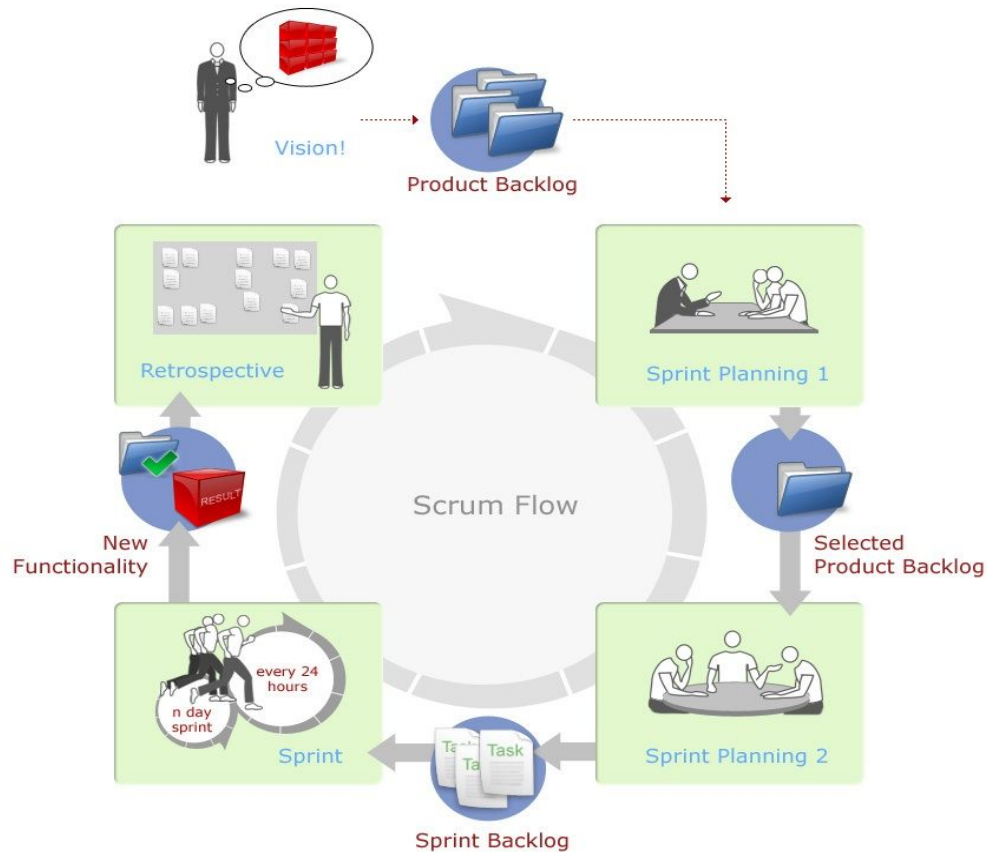
“Duas considerações:

- Sabíamos que, por serem “frameworks extensos”, estes modelos tendiam a gerar processos burocráticos, pelos mesmos motivos que implantações do Processo Unificado acabavam se transformando em Processos Cascata. Neste sentido, perdem seu mérito original... **não queríamos isso para a Powerlogic!**
- Através de livros, em 2006, percebíamos que a fundamentação básica do CMMI já vinha sendo interpretada de forma mais “ágil” por consultores e certificadores, **conscientes das burocracias desnecessárias advindas de aplicação de “fórmulas” não contextualizadas** (muitas advindas do Processo Unificado). **Este foi um grande motivador!”**



Por quê Scrum?

“O Scrum foi o framework de processo ágil com o qual nos identificamos, pela sua **visão gerencial pragmática** (complementar a práticas de construção do XP), com práticas claras visando objetivos bem fundamentados”



A vertical stack of software products from Powerlogic, connected by a downward arrow. From top to bottom:

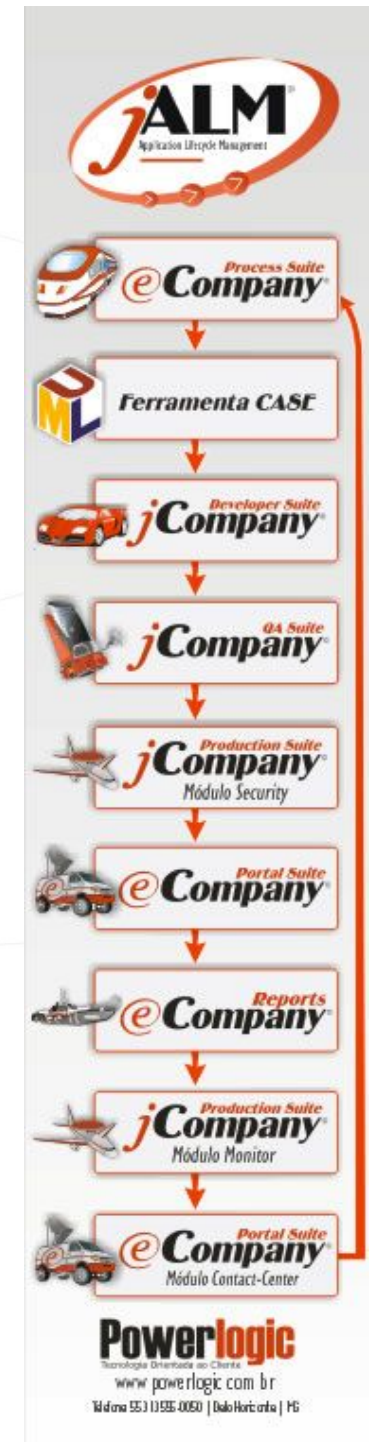
- jALM** Application Lifecycle Management
- @Company** Process Suite
- Ferramenta CASE**
- jCompany** Developer Suite
- jCompany** QA Suite
- jCompany** Production Suite Módulo Security
- @Company** Portal Suite
- @Company** Reports
- jCompany** Production Suite Módulo Monitor
- @Company** Portal Suite Módulo Contact-Center

 At the bottom of the stack is the **Powerlogic** logo and contact information:

Powerlogic
Tecnologia Orientada ao Cliente
www.powerlogic.com.br
Tel: (51) 31535-0050 | Dado Horário: | PG

Por quê Scrum?

“O Processo de Desenvolvimento de Produtos da Powerlogic traz ainda práticas extraídas do XP, PMI, UP e outras próprias. Porém, ao utilizar o Scrum como pilar, **enfatizamos o aspecto do “Iterativo” (Sprints) do processo, além de práticas de comunicação “banda larga”, com colaboração e pressão (peer pressure) em níveis apropriados...**”

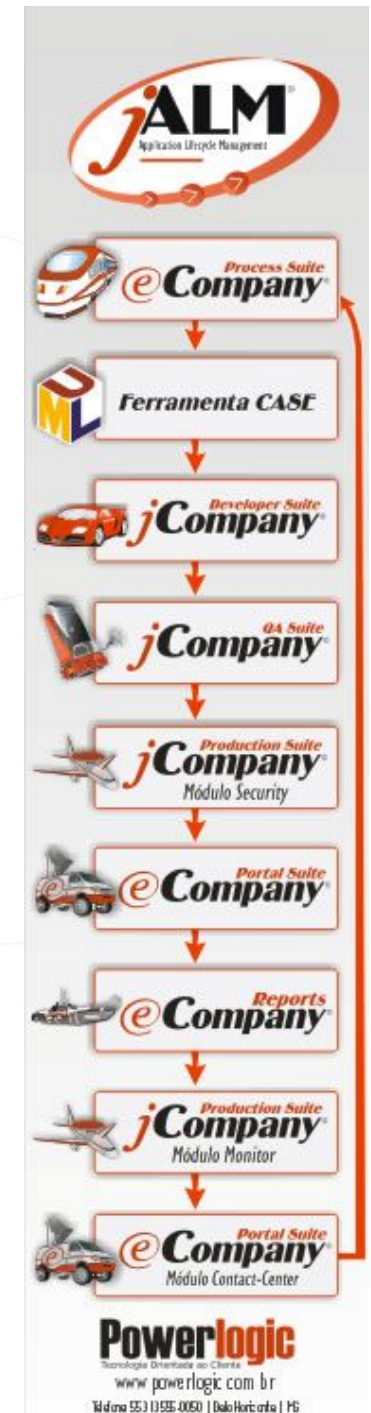


O Processo

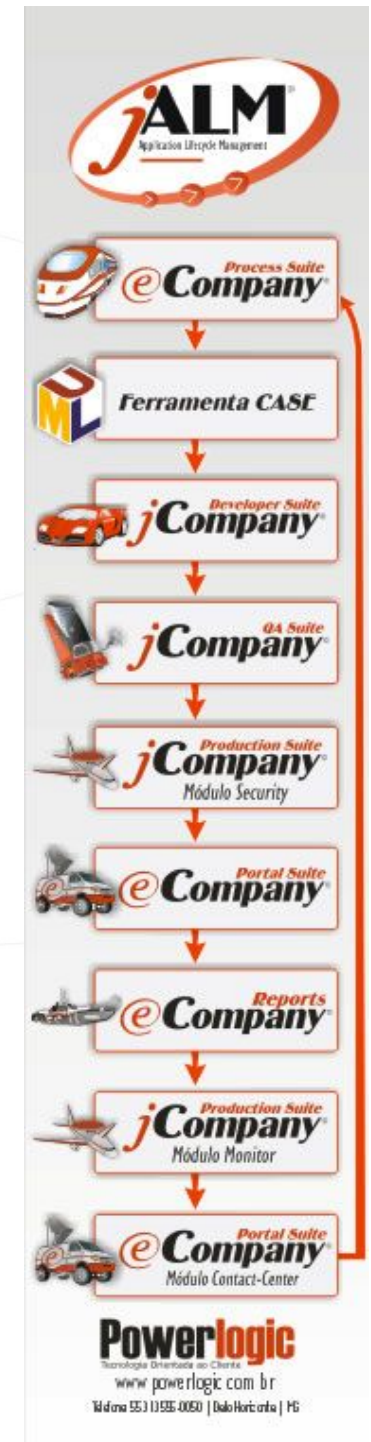
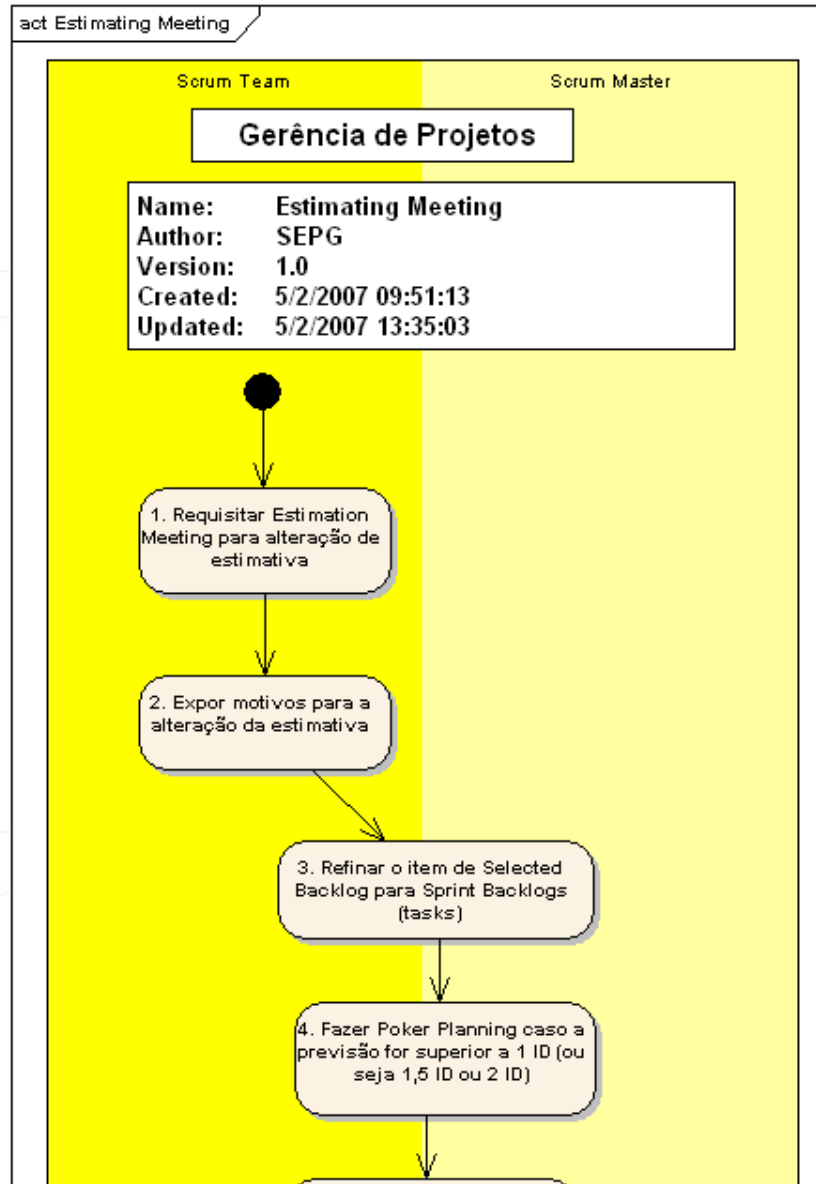


- ❑ **Pre-game:** Concepção arquitetural e organização inicial de um novo “release” (projeto). É como uma fase de “Concepção” mais curta de duração fixa (Time-Boxed), normalmente entre 7 a 15 dias.
- ❑ **Development:** Onde o principal acontece! Divide-se em Sprints (Ciclos de 30 dias). Um Release típico na Powerlogic é formado por 4 a 6 Sprints.
- ❑ **Monitoramento e Controle:** Atividades de “Medição e Análise” (MED) e de Gestão que averigam as atividades produtivas.
- ❑ **Post-Game:** Entrega final do produto (embalagem, documentação, liberação para o mercado)

- ❑ **Duração:** de maio/2006 até junho/2007. **Custo:** Aprox. R\$ 240Mil



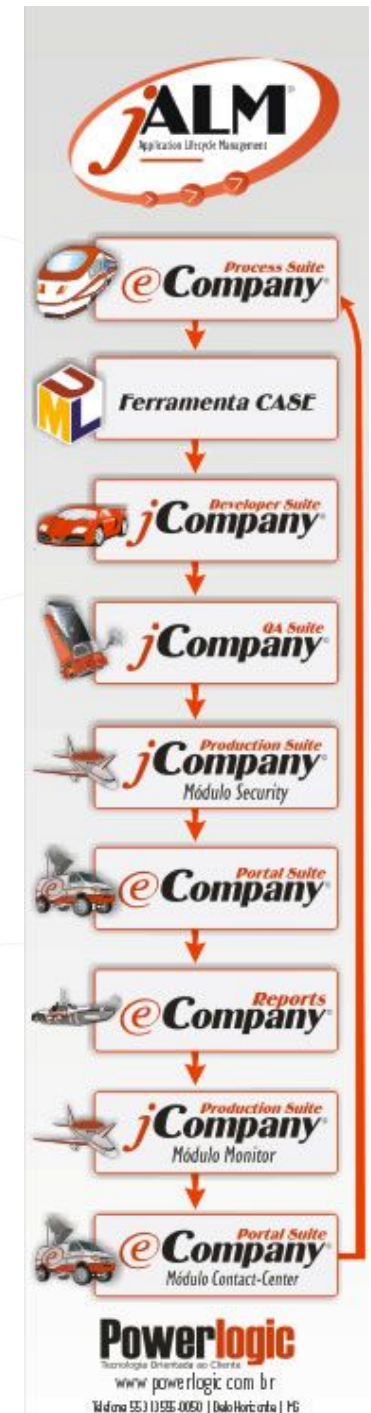
Representação do Processo



GPR – Gerência de Projetos

Alguns pontos chave:

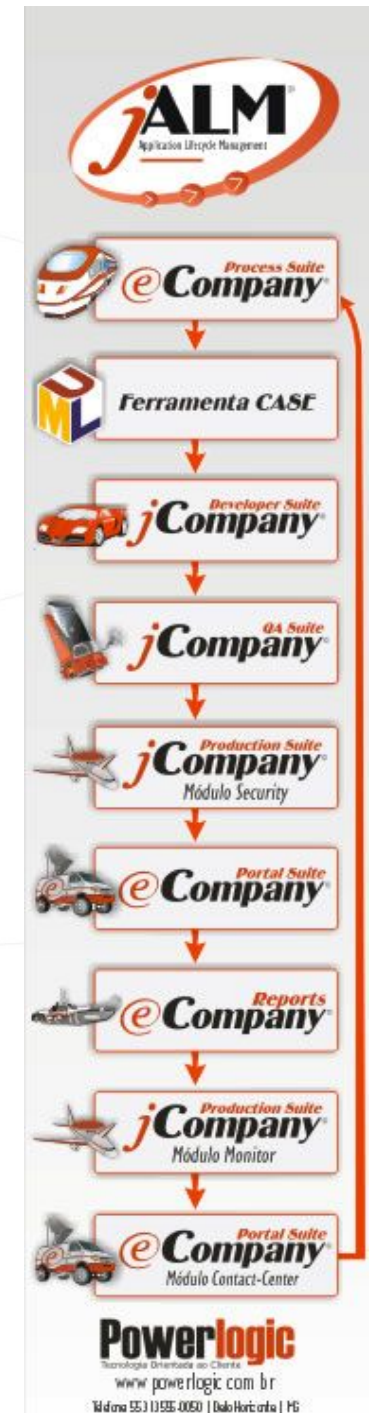
- ❑ Conceito de Release e Release Plan
- ❑ Planejamento via Release Planning, Sprint Planning 1 e Sprint Planning 2
- ❑ Acompanhamento contínuo via inspeção direta (Scrum Master)
- ❑ Acompanhamento “formal” diário via Daily Scrum, e mensal via Sprint Review e Release Review
- ❑ Agile Radiator com 3 bandas principais (Pendentes, Em Desenvolvimento e Finalizados), duas bandas para WWW e WCBI e uma banda para impedimentos.
- ❑ Reservas de “tempo para impedimentos” no planejamento de cada ciclo, revisada em cada Sprint Planning
- ❑ Isolamento da equipe
- ❑ **Práticas “adicionais”**: Registro no Powerlogic jALM (Release, Sprints, Goals, resultados,...); apropriação de horas diária; medição de produtividade individual (sendo questionada); matriz de competências.



GRE – Gerência de Requisitos

Alguns pontos chave:

- ❑ Uso de **Ideal Day** (considerando um “Ideal Man!”)
- ❑ Uso de Pocker Planning para Backlogs acima de 1 ID ou polêmicos
- ❑ Quebra de Product Backlog em Atividades que possam ser completadas por cada membro de equipe em 1 dia real.
- ❑ Captura contínua na forma de Product Backlog (eCompany Process Suite), com refinamento em três fases (Release Backlog -> Select Backlog -> Sprint Backlog)
- ❑ Solicitação de Mudança “leve”: Mudanças ao final do Sprint ou dentro de um Sprint que não afetem o “Sprint Goal” **não requerem “Solicitação de Mudança”**. Somente são requeridas em situações críticas (Ex.: “não vamos cumprir o Goal”) ou para mudanças de configuração (Ex.: “alteração de versão de framework”)
- ❑ **Práticas “adicionais”**: registro de Backlogs e métricas no eCompany Process Suite; rastreabilidade “do requisito ao código” (matriz de rastreabilidade)



GQA – Garantia da Qualidade

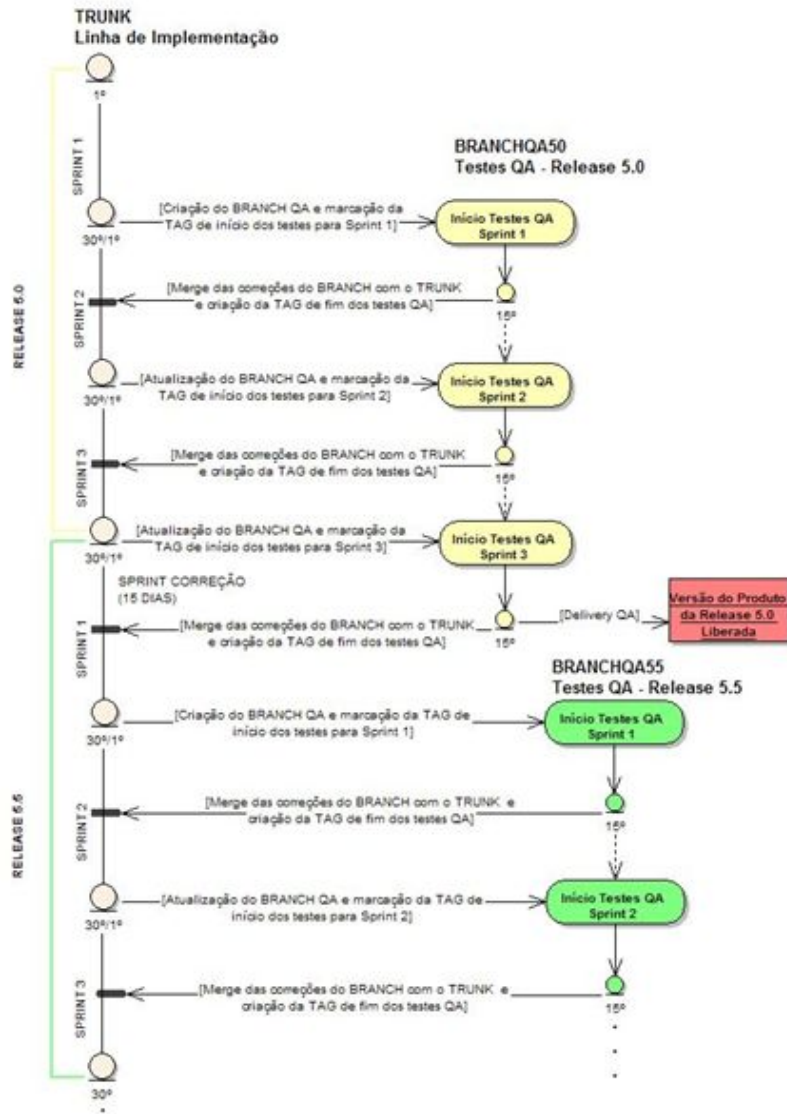


Alguns pontos chave:

- ❑ Introdução do conceito de **Post-Sprint de QA: Iterações de 15 dias (Time-Boxed) de QA, após cada Sprint. Iteração final pode exceder o tempo, em casos excepcionais.**
- ❑ **Conceito de “confirmação do Goal” pelo QA** (4 horas de Review não são suficientes para evitar que “não-conformidades” do Scrum Team cheguem ao mercado)
- ❑ Testes funcionais automatizados.
- ❑ Erros capturados pelo QA Team contam pontos contra o Scrum Team. Erros capturados pelo mercado contam pontos contra o QA Team.

A vertical stack of software products from Powerlogic. At the top is the **jALM** logo (Application Lifecycle Management). Below it are several product boxes: **eCompany Process Suite**, **Ferramenta CASE**, **jCompany Developer Suite**, **jCompany QA Suite**, **jCompany Production Suite Módulo Security**, **eCompany Portal Suite**, **eCompany Reports**, **jCompany Production Suite Módulo Monitor**, and **eCompany Portal Suite Módulo Contact-Center**. At the bottom is the **Powerlogic** logo and contact information: www.powerlogic.com.br, Tel: (11) 595-0050 | Belo Horizonte | MG.

GCO – Gerência de Configuração



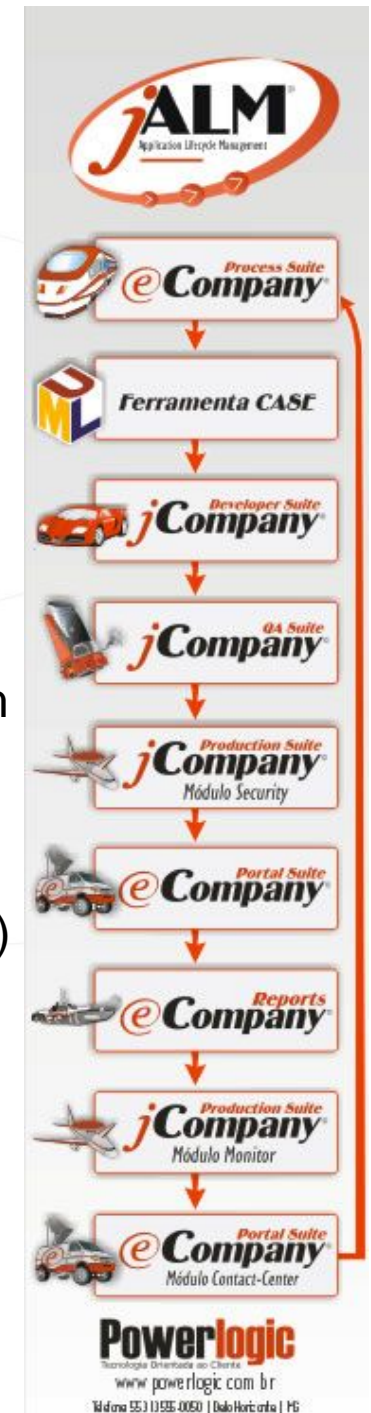
Alguns pontos chave:

- Integração Contínua (SVN, Maven e Continuum) com jCompany QA Suite
- Rastreabilidade do Requisito (Backlog) ao Código
- Controle de código fonte, componente (JAR), executável (WAR), release plan (com linha de base), documentação e mídia de CD/DVD dos produtos.



Alguns pontos chave:

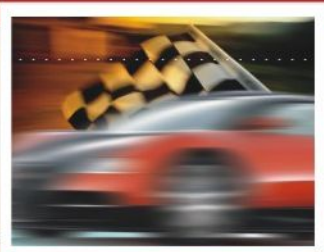
- ❑ Indicadores “Ágeis”: Assiduidade do Daily Scrum, Remoção de Impedimentos, Frequência de Integração, Inspeção,...
- ❑ Indicadores “Clássicos”: Produtividade (Velocidade*Qualidade), Metas (Goals), Previsto x Realizado, ...
- ❑ Resultados tangíveis são vistos positivamente pelo Scrum Team (produtividade da Scrum Team, do QA, individual?, etc.)
- ❑ Gerente de Processo como apoio ao Scrum Master no incentivo e catequese de práticas ágeis.
- ❑ Gerente de Processo como assessor da Diretoria (Management) para garantia de resultados da implantação do processo.





MPS.Br Nivel C (CMMI 3)

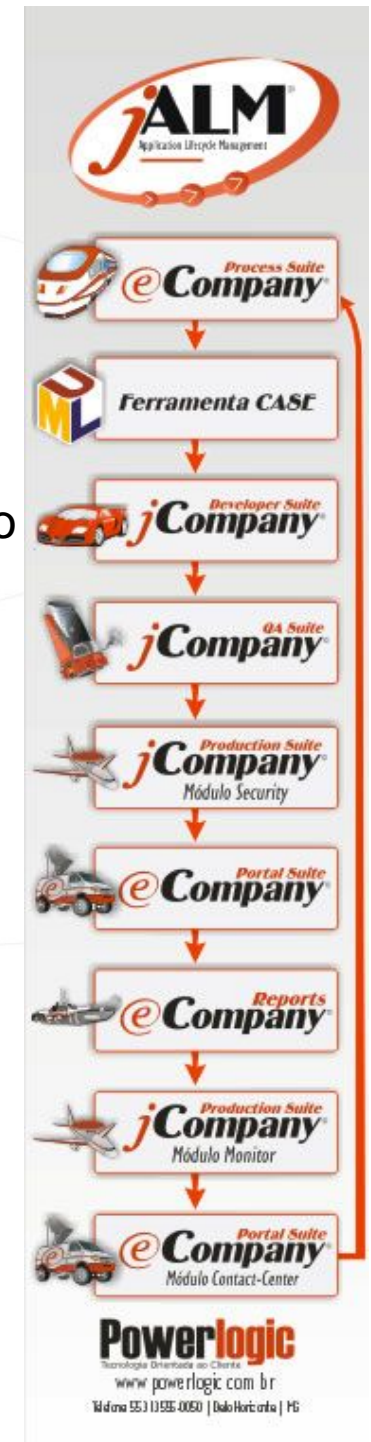
O que estamos vislumbrando (2008-2009)



MPS.Br Nivel C & Agile

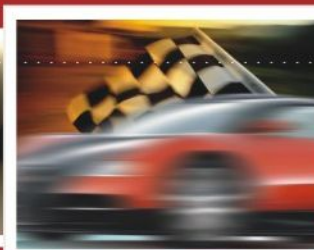
Alguns pontos chave:

- ❑ Evidências na Área de Engenharia de Software obtidas de práticas de XP (TDD, Pair-Programming, Refactoring, Integração Contínua) e formalização de alguns diagramas arquiteturais mínimos (CASE UML)
- ❑ Maior importância no uso de uma suíte de ALM para obtenção de evidências “por consequência do processo produtivo”, integrando ECM, Portal e CASE.
- ❑ Caracterização do espectro de “agilidade” para cada projeto (Escala de Cockburn + Matriz de Restrições)
- ❑ Expansão em áreas de gestão de RH (plano de treinamento, evolução individual, etc.), consoantes com os valores ágeis.
- ❑ ?





Obrigado



Paulo Alvim
(alvim@powerlogic.com.br)