

Projeto de Desenvolvimento de Sistemas

Projeto, Desenvolvimento, Sistema

- Conceitos, em geral, focados na construção de componentes de
 - software e hardware
- É comum abranger "descrição de rotinas"
 - usadas pelos utilizadores diretos de cada automatismo no sistema
- Não é comum abranger as “atividades circundantes”, que
 - usualmente se realizam durante a construção de modelos de negócio
- A abordagem à noção de “projeto” será ilustrada essencialmente
 - na perspectiva do desenvolvimento de software
 - ... mas aplica-se às várias disciplinas de projeto de base tecnológica
- ... a **ênfase** é a da necessidade de uma nova atitude ...
 - uma atitude industrial face ao desenvolvimento de tecnologia
 - ... em especial face ao desenvolvimento de sistemas de *software*!

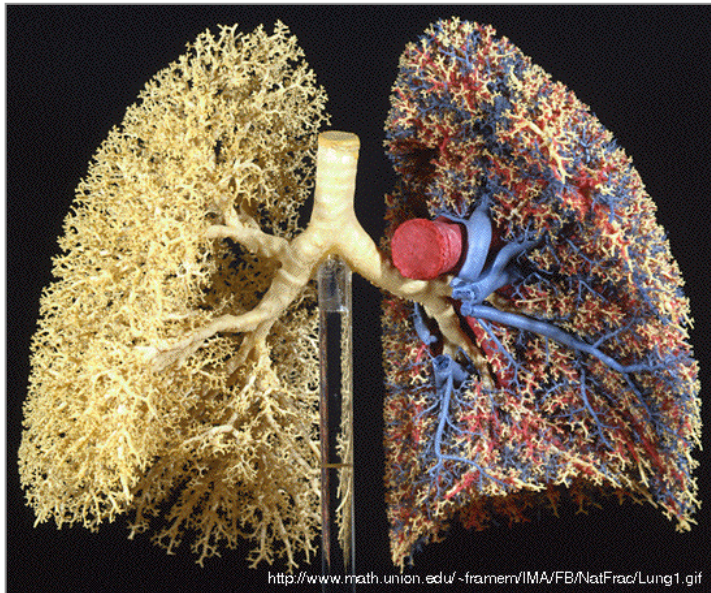
Atitude Industrial *versus* Atitude Artesanal

- Industrial – relativo à indústria
 - “atividade económica que se utiliza de uma técnica, dominada, em geral, pela presença de máquinas ou maquinismos, para transformar matérias-primas em artefactos acabados”; *cf. dicionário Porto Editora*
- Artesanal – fabricado por artesão
 - “artífice em que coincidem a posse dos meios de produção, o fornecimento da força de trabalho e o contacto direto com o mercado”; *cf. dicionário Porto Editora*
- Ambiente Industrial – exemplo de produtos
 - automóvel, casa, ponte, navio, computador, servidor e-mail, Java, ...
- Ambiente Artesanal – exemplo de produtos construídos em
 - obra de arte, comida caseira, exercício de programação em aula, ...

Atitude Industrial e Produto Complexo

... é complexo!

... mas tem alguma relação com a “atitude industrial”?



<http://www.math.union.edu/~framem/IMA/FB/NatFrac/Lung1.gif>

A natureza “fabrica os artefactos” mais complexos.

... mas o “processo de fabrico” da natureza demora demasiado tempo!

... e desconhecemos o detalhe do processo de fabrico!

... e desconhecemos como garantir a manutenção dos artefactos!

Precisamos de fabricar artefactos “também complexos” mas em tempo útil com processos bem-definidos e garantir a sua manutenção ao longo do tempo.

Produto complexo? – “Casota de Cão”

Tantas madeiras!
Tantos pregos!
Martelo, Serrote!
Fita Métrica...
➔ é complexo!



Paulo Trigo Silva

Desenho mínimo
Ferramentas simples
Tempo e paciência!
➔ não é complexo!



Produto complexo? – “Moradia”

Desenho de arquiteto
Várias especialidades
Trabalho em equipa
Tempo e muito dinheiro!
➔ é complexo!



Paulo Trigo Silva



Arquitetura simples
Processo não muito longo
Pequenas equipas
Tempo e custo controlado!
➔ não é complexo!

Produto complexo? – “Arranha-Céus”

Arquitetura sofisticada

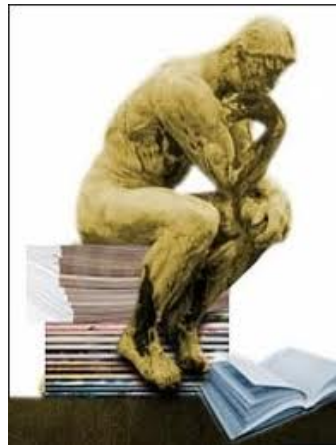
Processo muito longo

Imensas equipas

Tempo e custo enormes!

Explora limite tecnologia e conhecimento

→ é complexo!



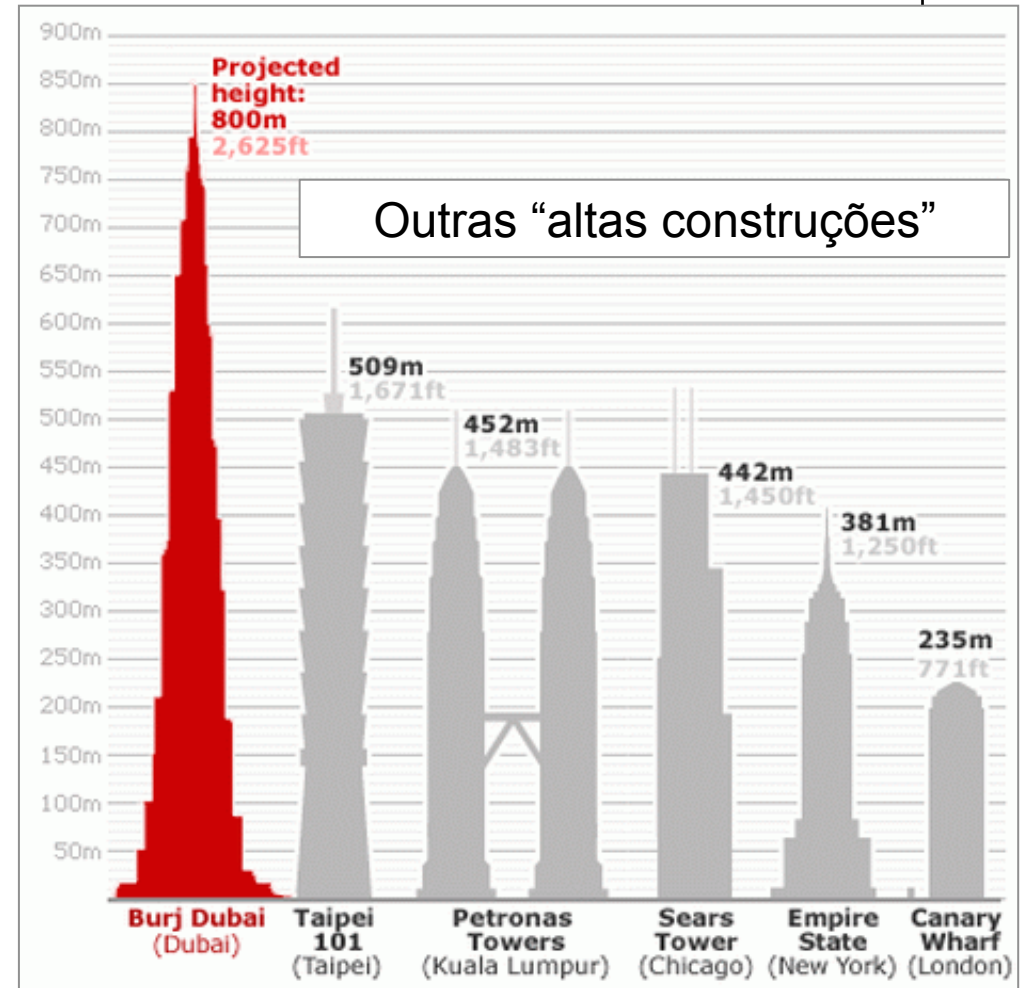
Burj Dubai
± 800m



“Arranha-Céus” – algumas curiosidades



Burj Dubai ± 800m; 160 andares
Mais de 12 mil operários trabalharam
na construção do Burj Dubai que tem
57 elevadores e três mil vagas de
estacionamento subterrâneo.

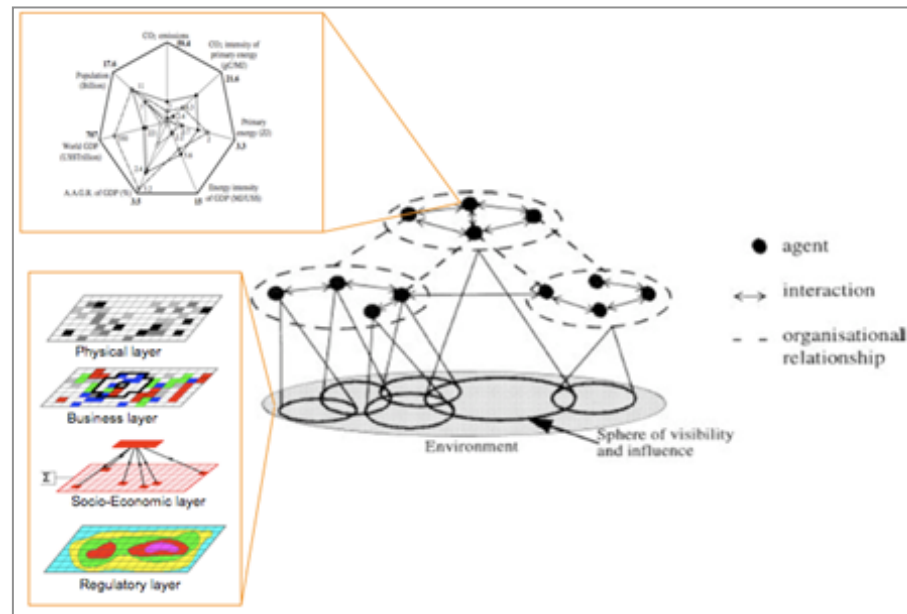
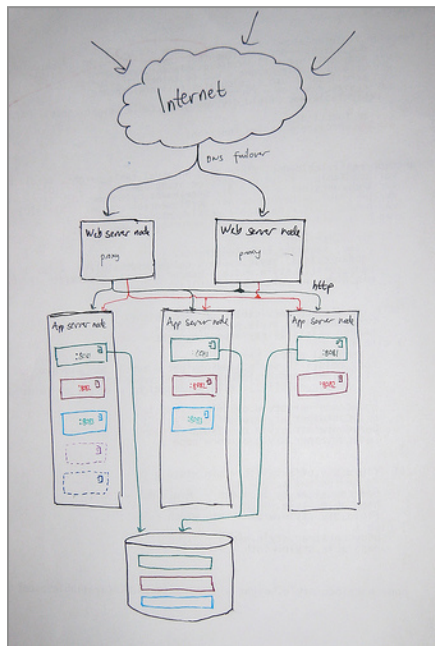


4.Jan.2010

Produto complexo? – “software”

Tendência para apoiar o homem na tomada de decisão, simular fenómenos naturais, organizar dados e extrair informação, simplificar comunicação e transferência de conhecimento, fornecer momentos lúdicos com elevado realismo, controlar outros sistemas (também complexos?),

Exige processo bem-definido, várias especialidades, especificação formal.



Produtos complexos e seu “processo de fabrico”

- Num produto complexo são essenciais as "descrições de rotinas"
 - para desenvolvimento e posterior manutenção contínua do produto
 - e.g., o plano de revisões de um automóvel ...
- As “rotinas” são parte integrante do desenvolvimento do produto
 - esta atitude está enraizada nos diversos sectores industriais
- Sistemas de software, são produtos complexos
 - até controlam atividades de produção de outros produtos industriais!
- O software também necessita de "descrições de rotinas"
 - que venham a ser o suporte da sua constante manutenção
- Para construir, manter e reparar produtos, como um todo,
 - o software deve ser encarado como qualquer outra peça do sistema ...
 - até porque por vezes é um produto que controla a produção de outros!

Algumas características do software

produção de software deve adoptar atitude industrial

software is developed or engineered, it is not manufactured in the classical sense

produção do software é diferente da de hardware

software doesn't "wear out"

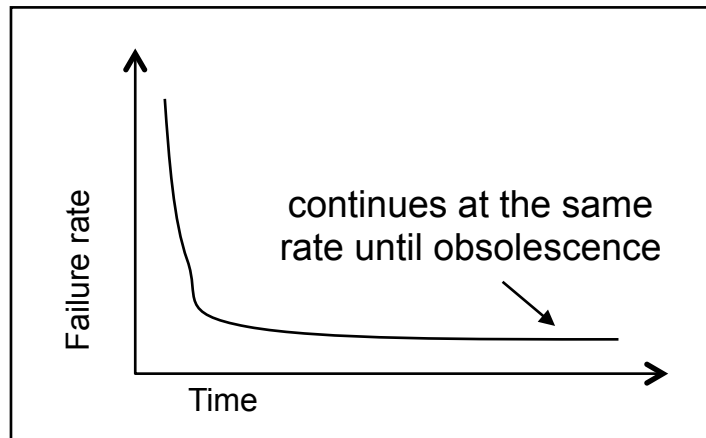
reutilização no software precisa de ser ainda mais efetiva

most software is custom-built, rather than being assembled from existing components

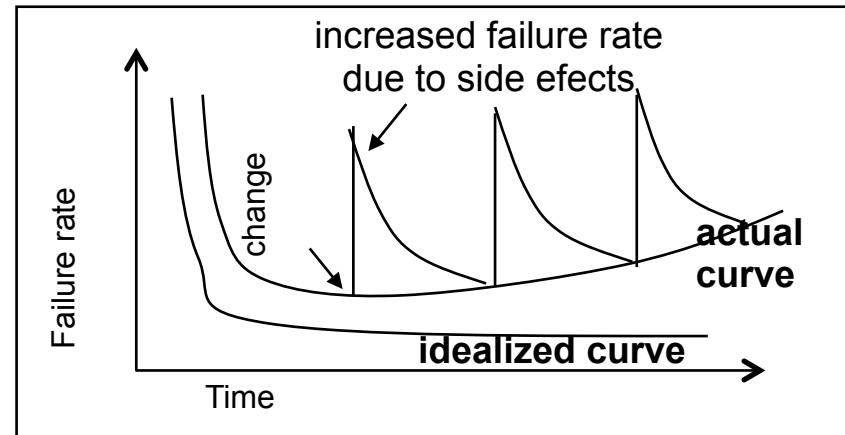
in Roger S. Pressman; *Software Engineering*; p. 10-12; 4th edition

O “efeito do tempo” sobre o software (e hardware)

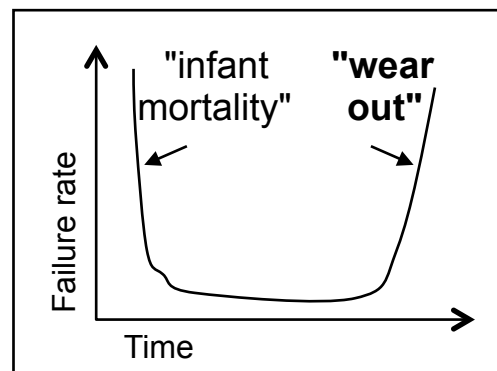
Failure curve for software (idealized)



Actual failure curve for software



Failure curve for hardware



A produção de software é diferente?

- É uma indústria recente – algumas décadas (1970-C; 1980-SQL)
 - a indústria automóvel tem mais de um século ... (1900, FordT)
 - automóvel tem essencialmente "mesmo aspecto" há cerca de 50 anos
 - ... mas um longo tempo se passou até se obter um consenso em relação à existência de 4 rodas, 1 volante, 1 acelerador, ...
- Um programa é uma entidade abstracta
 - exige, das pessoas e ferramentas, forte capacidade de abstração
 - é mais simples falar de objetos concretos – relógio, computador, ...

produção de software deve adoptar atitude industrial:

*... obviously, software must be developed on the same condition
as other types of industrial products*

in Ivar Jacobson; *Unified Software Development Process*

O software e os “mitos” – ao nível da Gestão

- Tenho livro de diretivas e procedimentos para construir software.
 - o livro é usado pela equipa ou é “peso burocrático”?
 - a equipa conhece e revê-se nas práticas aconselhadas no livro?
 - o livro reflete técnicas modernas de desenvolvimento de software?
- A minha equipa tem o mais recente equipamento informático.
 - o último modelo de hardware não é suficiente
 - na comunicação, a especificação sem ambiguidade (formal) ajuda mais
 - na produtividade, e qualidade, as ferramentas CASE ajudam mais
- Se o desenvolvimento atrasar, contrato mais elementos para a equipa e acerto no prazo (“Mongolian horde concept”).
 - novos elementos exigem formação
 - criam-se novos canais de comunicação
 - ... logo, reduz o tempo despendido em esforço produtivo e, em geral, contribui para atrasar ainda mais o projeto e torná-lo bem mais caro

... combater os “mitos” – ao nível da Gestão

Adoptar um processo de desenvolvimento testado e aceite pela comunidade como espelhando as “boas práticas de Engenharia”;

e.g., “Rational Unified Process”, “Srcum” ou mistura deles

Comunicar sem ambiguidade usando linguagens formais (mas intuitivas) e que ajudem a seguir o processo de desenvolvimento adoptado;

e.g., “Unified Modeling Language” e ferramenta “StarUML”

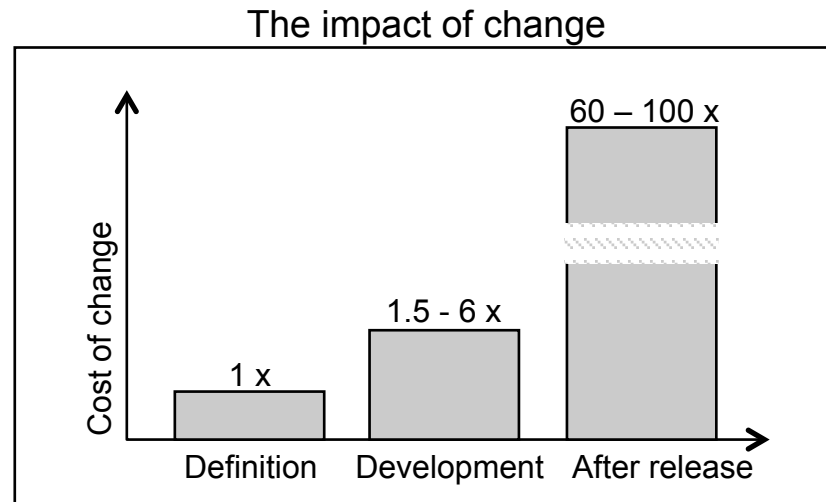
Identificar e planear tarefas considerando possíveis riscos, períodos de menor disponibilidade de tempo e momentos de entrega de artefactos;

e.g., “Escalonamento de tarefas e recursos” com ferramenta “GanttProject”

Escolher projeto e equipa de trabalho motivada nos temas do projeto.

O software e os “mitos” – ao nível do Cliente

- Uma definição geral de requisitos é suficiente para se começar o desenvolvimento. Depois logo se definem os "detalhes".
 - fraca definição de requisitos é a maior causa de falha dos projetos
- Os requisitos de projeto mudam frequentemente, mas as mudanças podem ser incorporadas de modo simples pois o software é flexível.
 - os custos das mudanças de requisitos variam com a fase do projeto
 - após definição de recursos e metodologias os custos sobem muito



... combater os “mitos” – ao nível do Cliente

Fazer um levantamento de requisitos que não seja superficial mas tendo também em atenção a necessidade de delimitar o âmbito do projeto.

Reduzir o risco fazendo análise de outras soluções na área e solicitando prototipagem rápida para explorar os requisitos que estejam menos claros.

Definir marcos para entrega de artefactos e alinhar pagamento de facturas com a aceitação dos artefactos produzidos pelos fornecedores.

Seguir de perto a execução do projeto e ter capacidade de resposta.

O software os "mitos" – ao nível do Especialista

- Assim que o programa funcionar sem erros o trabalho está feito.
 - *"the sooner you begin writing code, the longer it'll take you to get done"*
 - ... 50 a 70 % do esforço investido num programa será despendido após entrega, ao cliente, da sua primeira versão; a consultoria intervém aqui!
- Só "em produção" posso aferir a qualidade do programa
 - as revisões técnicas ao software podem ser feitas desde o primeiro momento e são um mecanismo eficaz para aferição de qualidade
- O único artefacto num projeto de sucesso é o seu código.
 - programa em funcionamento é só parte do que é necessário entregar
 - a documentação descreve o essencial das ideias vertidas no código
 - a documentação constitui a base do sucesso no desenvolvimento; contém as linhas de orientação para as atividades de manutenção
 - a formação é a base para o cliente ganhar segurança e usar o sistema

... combater os “mitos” – ao nível do Especialista

Planear o esforço considerando as várias fases (e iterações) do processo de desenvolvimento adoptado; e.g., desenvolvimento, teste, manutenção.

Programar fazendo testes à medida que o código evolui e solicitar que outros elementos da equipa façam testes sobre o que eu implemento.

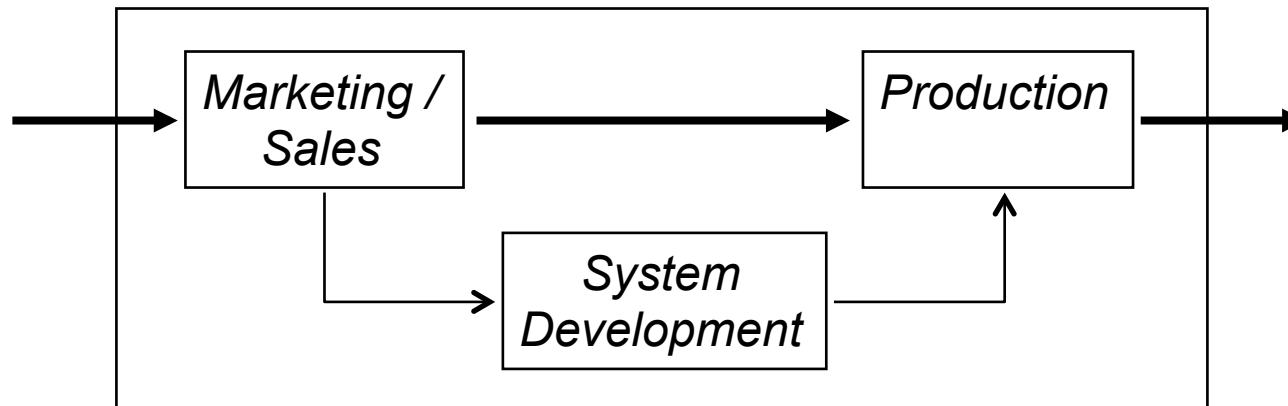
Elaborar documentação à medida que o projeto evolui e fazer modelos para partilhar com outros, e consigo mesmo, as opções de implementação.

Garantir que o sistema desenvolvido é testado e aceite pelo cliente.

Modelar antes de codificar, testar o código e documentar as ideias.

Desenvolvimento de Sistema de Software (Aplicação)

- É usualmente parte de uma atividade mais abrangente
 - em alguns casos, a atividade visa construir um produto do qual o sistema de software é parte integrante
 - o caso de grandes indústrias, como a automóvel
 - noutros casos, a atividade pode abranger o departamento de processamento de dados de uma empresa
 - o caso de companhias de seguros e bancos
- ... parte de uma atividade mais abrangente ...



... foco no Desenvolvimento do Sistema

- Objectivo
 - partir de um conjunto de requisitos novos (ou alterados)
 - construir um sistema novo (ou alterado)
- Desenvolvimento de Sistemas pode ser visto como
 - um processo e suas especificações para produção
- Esta visão percorre todas as atividades
 - análise, desenho, programação e teste
- Neste contexto, o código é uma especificação,
 - que pode ser entendido por programadores e
 - pelo processo de produção – pelo compilador

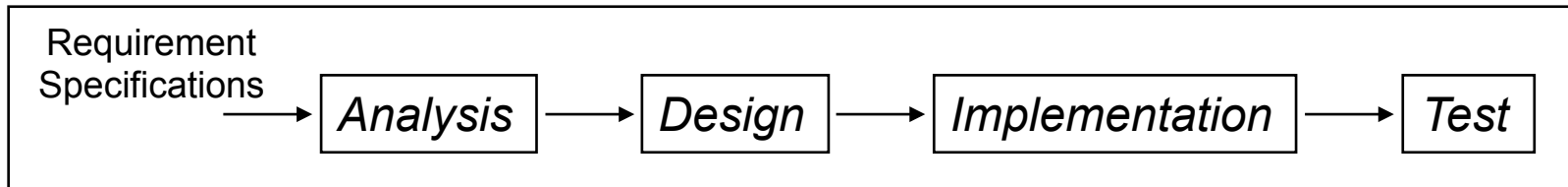


Especificações

- São descritas através de modelos
 - com diferentes níveis de detalhe
- Os primeiros modelos são muito abstractos e identificam
 - aspectos funcionais e não-funcionais do sistema
 - ... aspectos funcionais – os serviços fornecidos aos utilizadores
 - ... aspectos não-funcionais – suporte tecnológico dos serviços
- Os modelos finais tornam-se mais detalhados, descrevem,
 - como o sistema será construído, e
 - como irá funcionar
- A existência de diferentes modelos sugere a existência de
 - diferentes atividades na construção de modelos
 - diferentes fases no refinamento dos modelos

Atividades de construção de modelos

As atividades – simplificadas e numa evolução do "tipo cascata"



Cada atividade acrescenta à anterior:

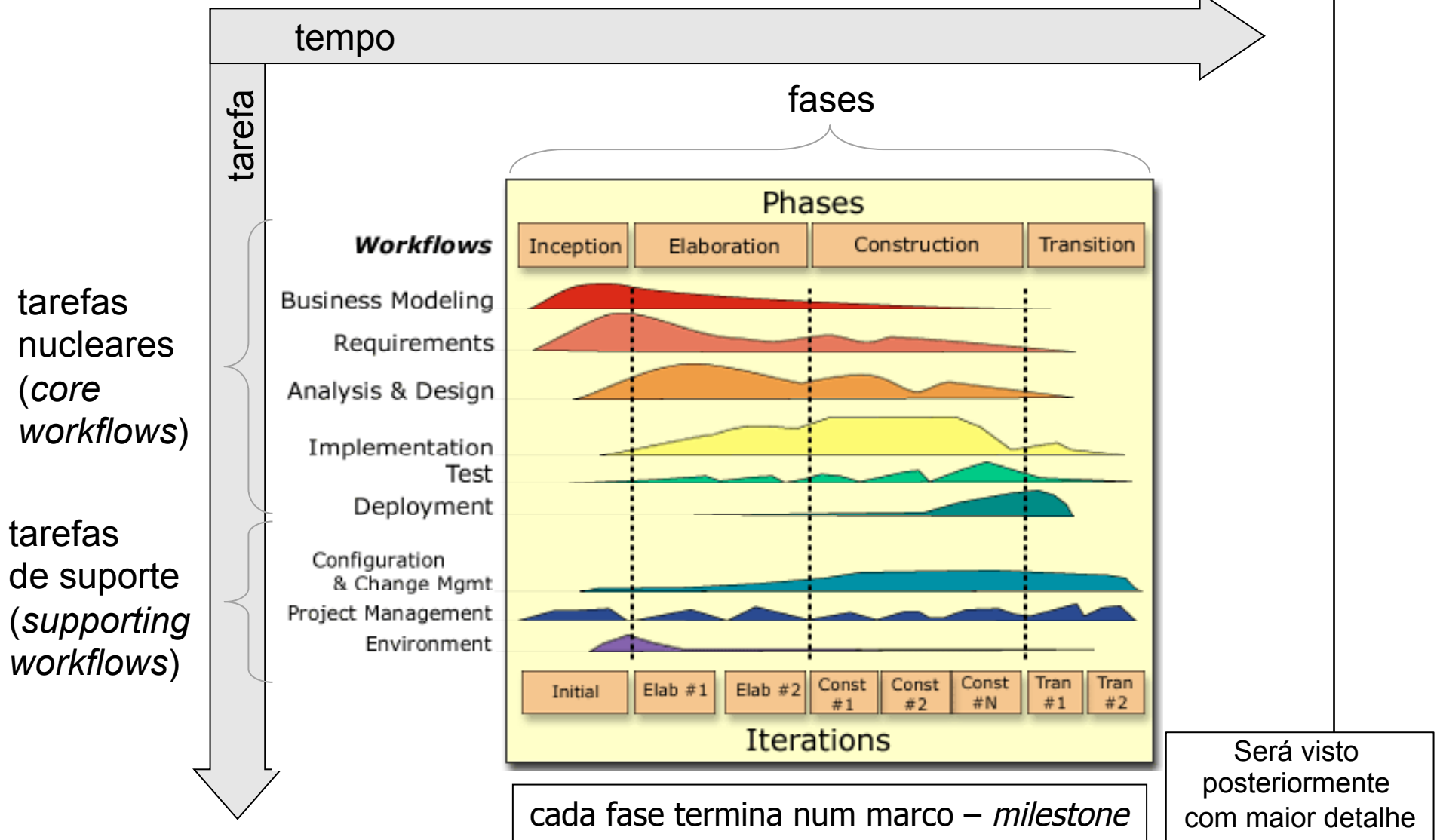
- detalhe na especificação

e.g., desde a noção geral de personagem até à definição das suas características gráficas e capacidade de visão artificial

- rigor no formalismo

e.g., desde descrição textual até codificação Java&SQL&OpenCV

... atividades com evolução “não-cascata” (incremental)



Atividades de construção de modelos – Análise

- Análise (*Analysis*)
 - especifica o que (***what***) o sistema vai oferecer aos utilizadores
 - define uma estrutura "robusta" a alterações
- ... constrói arrumação clara em unidades / serviços
 - que "se podem adquirir"
 - define "arquitetura candidata"
 - ... adopta fracos compromissos tecnológicos
- Modelo de Análise (*Analysis Model*)
 - especificação resultante da atividade de Análise
 - esboço do comportamento do sistema em condições ideais
 - independente de um qualquer ambiente de implementação

Atividades de construção de modelos – Desenho

- Desenho (*Design*)
 - especifica como (**how**) concretizar o modelo de análise
 - e.g., usando SGBDs, interfaces pessoa-máquina, ...
- ... gradualmente substitui as condições ideais de análise
 - por requisitos derivados do ambiente de implementação adotado
 - assume compromissos com tecnologias e arquiteturas
 - ... e.g., Java, PostgreSQL, Apache Tomcat WebServer
- Modelo de Desenho (*Design Model*)
 - formalização do modelo de análise
 - agora considerando o ambiente de implementação
 - refinando a arquitetura candidata face às tecnologias adotadas

Atividades de construção de modelos – Implementação

- Implementação (*Implementation*)
 - concretiza padrões definidos nos modelos de desenho
 - constrói interfaces, classes, e relações entre classes
 - codifica algoritmos identificados na atividade de desenho
 - garante persistência dos dados e comunicação entre processos
 - concretiza interação homem-máquina
- ... evolui a partir das opções de desenho
 - mas também contribui para a revisão desses mesmas opções
 - ... a codificação pode originar ajustes nos modelos de desenho
- Modelo de Implementação (*Implementation Model*)
 - refinamento do modelo de desenho
 - ... de modo a melhor explicar o essencial do código desenvolvido

Atividades de construção de modelos – Teste

- Teste (*Test*)
 - encontrada a realização operacional do modelo de análise?
 - tudo está corretamente implementado & desempenho é aceitável?
- Vários níveis de Teste
 - ao longo do desenvolvimento
 - em ambiente de pré-produção
 - em conjunto com os utilizadores (testes de aceitação)
 - em ambiente de produção durante período de pré-implantação
- Modelo de Teste (*Test Model*)
 - sequência de ações e respetivo resultado esperado (causa-efeito)
 - pode ser especificado por equipa de utilizadores (ou de “testadores”)
 - devem ser construídos “scripts” que automatizem testes especificados
 - todos os testes devem ser executados após qualquer alteração

Atividades de construção de modelos – Síntese

- Cada atividade acrescenta ao resultado da anterior
 - detalhe na especificação
 - rigor no formalismo
- O sistema "nasce e vai crescendo" com as diferentes atividades
 - até se atingir o último nível de detalhe – o código do programa
- O último nível de detalhe não significa o "fim do sistema"
 - espera-se que o sistema dure para além da sua primeira versão!
- "Mudança e Revisão" constituem grande parte da fatia dos custos
 - inerentes ao ciclo de vida de um sistema; consultoria atua aqui!
- Normalmente o sistema evolui ao longo de várias versões
 - ... uma atitude industrial tem que considerar esta perspectiva ...

... da Análise ao Desenho

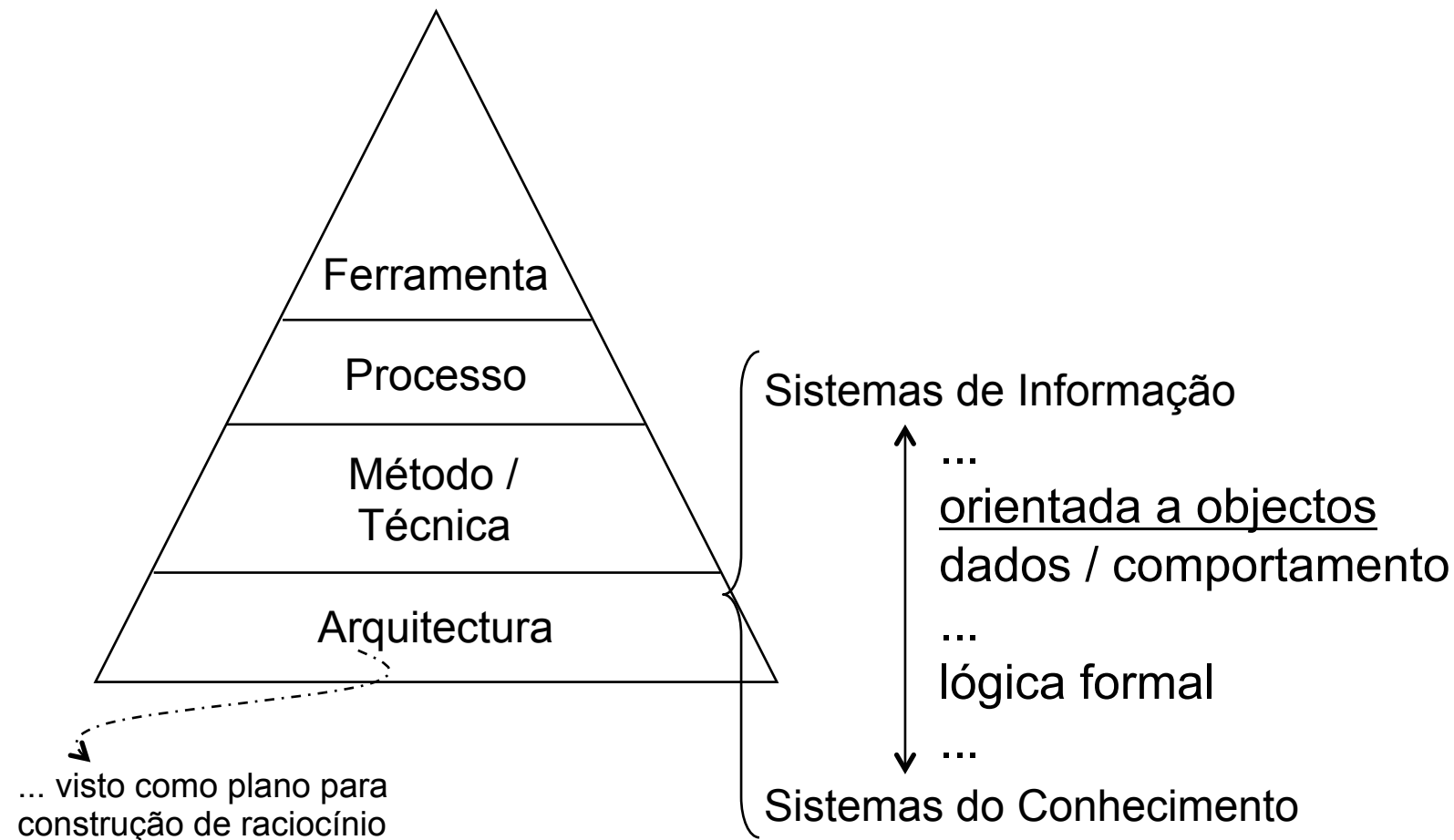
- Em qualquer metodologia de desenvolvimento é possível
 - traçar, de diversos modos, fronteiras entre as diferentes atividades
- As fronteiras escolhidas podem variar de projeto para projeto
 - a análise pode atingir alto nível de detalhe
 - ... e.g., quando serve de base ao orçamento (de um fabricante)
- Análise poderá estar muito fracamente ligada à implementação
 - se for evidente o modo como as restrições impostas pelo ambiente de implementação influenciam os requisitos
 - ... e.g., a definição de “satisfatório nível de conforto”, pode ser uma especificação de análise de um sistema de regulação de temperatura de um aparelho de ar condicionado
- Para a análise não existe ainda uma técnica estritamente formal
 - é comum usar técnicas descritivas, práticas e simples de entender
 - ... iremos considerar especificação via “casos de uso” (*use cases*)

... da Análise ao Desenho – linhas de orientação

- Modelo de Análise descreve um "mundo ideal"
 - "relaxar" restrições de memória, desempenho, tolerância a falhas
- Modelo de Análise descreve o sistema em termos de conceitos
 - os conceitos são partes integrantes do sistema (e.g., serviços)
 - ... é o suporte do desenho, pelo que a estrutura de implementação irá espelhar a estrutura do problema e não ao contrário ...
- Modelo de Análise independente do ambiente de implementação
 - alterações nos requisitos de implementação não afectam a análise
 - e.g., mudança do SGBD não implica alterar o modelo de análise
- Modelo de Análise não deve ter detalhe a adaptar
 - não elaborar detalhe a alterar devido ao ambiente de implementação
 - ... pode ocorrer se o modelo de análise for demasiado formal

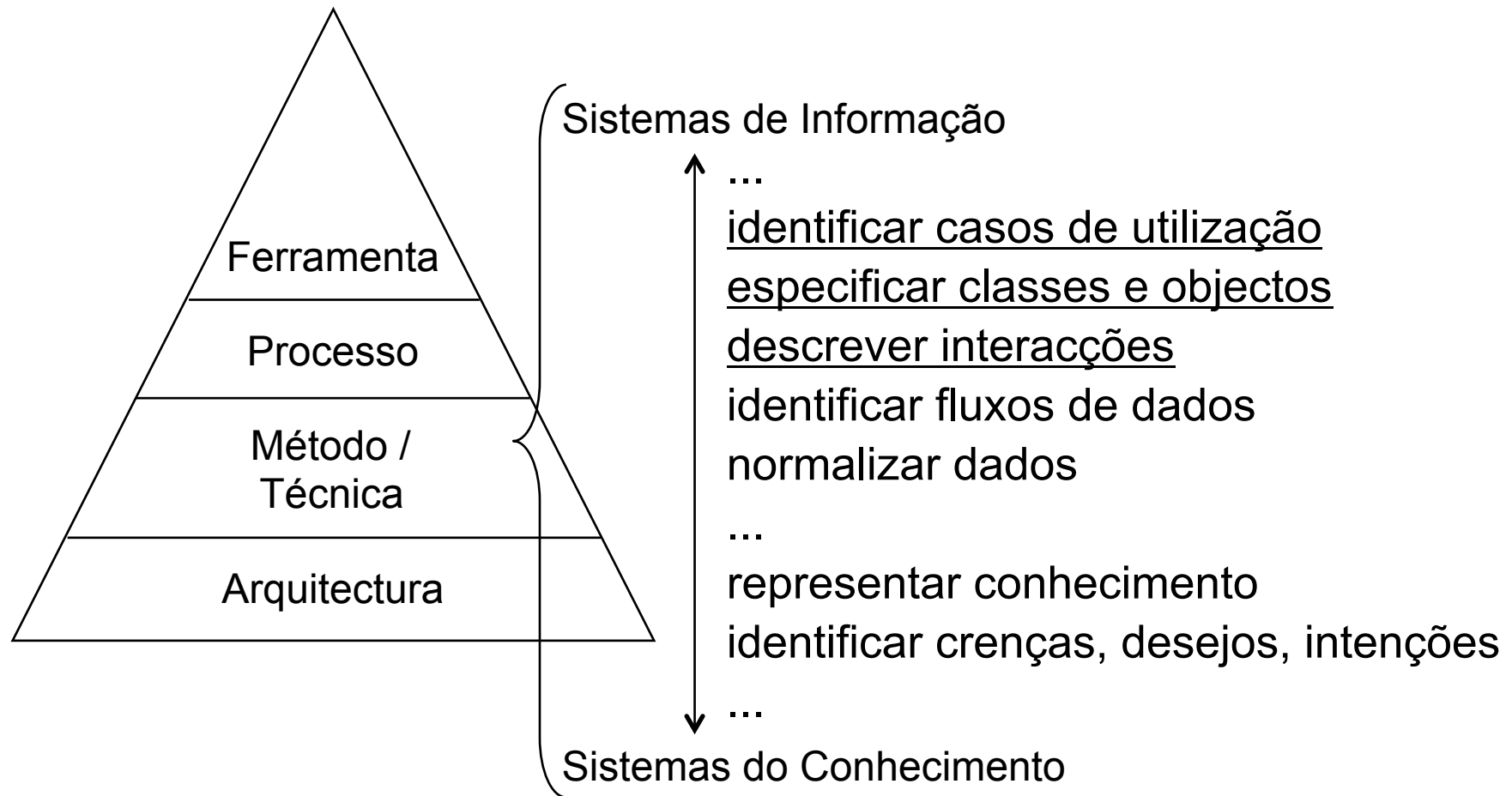
Atitude Industrial – Arquitetura

- Os blocos que contribuem para esta atitude ...



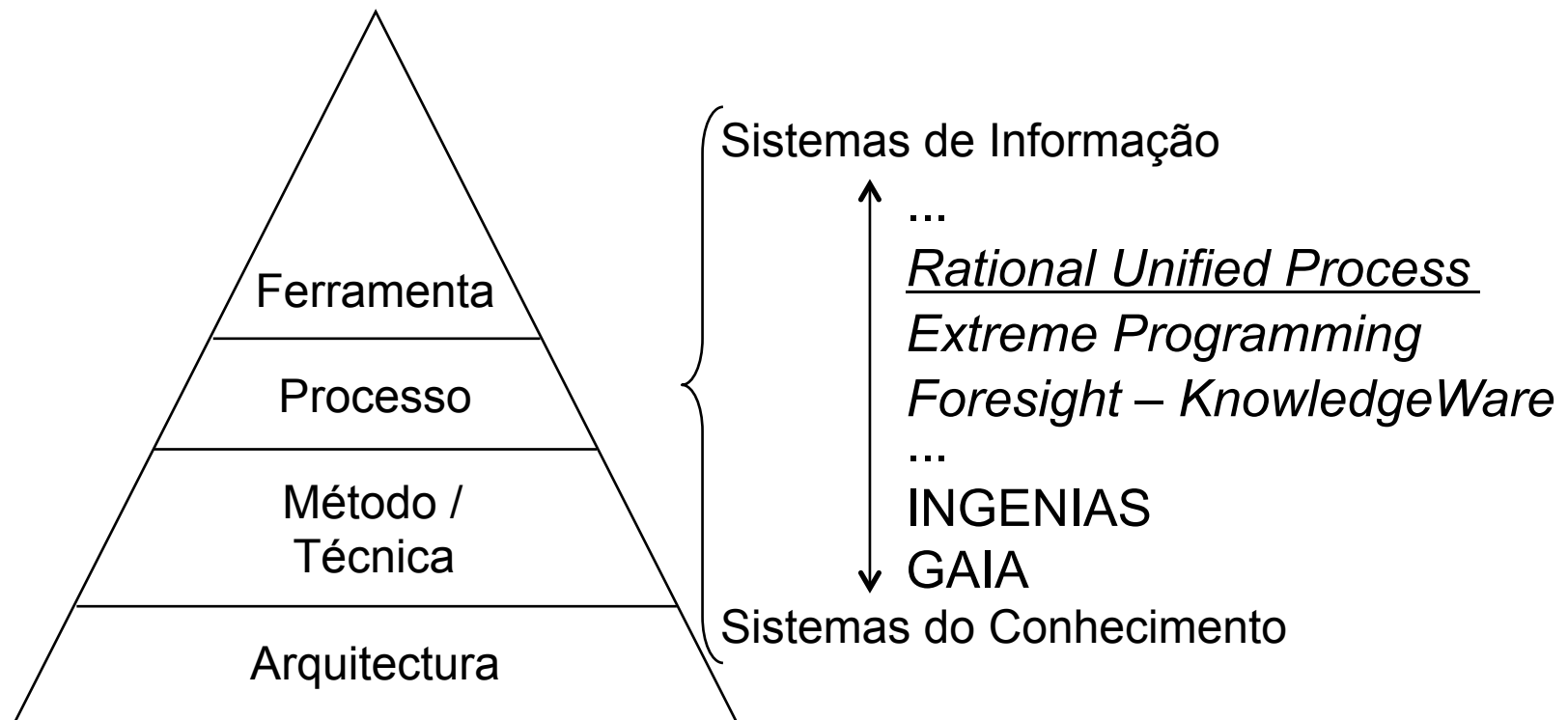
Atitude Industrial – Método/Técnica

- ... os blocos que contribuem para esta atitude ...



Atitude Industrial – Processo

- ... os blocos que contribuem para esta atitude ...



Atitude Industrial – Ferramenta

- ... os blocos que contribuem para esta atitude ...

