

<http://codinghorror.typepad.com>

# COLABORAÇÃO EM PROJETOS DE SOFTWARE

# Trabalho em Equipa



- Não trabalhar em equipa normalmente implica:
  - ▣ Despender grande esforço
  - ▣ Elaborar algo simples e limitado
  - ▣ Não cumprir com metas temporais esperadas
  - ▣ Criar algo não robusto ao mundo real
    - Visão própria/falta de informação
  - ▣ ...

# Trabalho em Equipa



Projetos são “**sempre**” realizados por equipas

# Trabalho em Equipa



- Saber trabalhar em equipa é uma das características principais no mundo atual!
  
- Exemplo do sítio <http://www.itjobs.pt>
  - ▣ Palavra "equipa": 2026 propostas
  - ▣ Palavra "programação": 335
  - ▣ Palavra "java": 735
  - ▣ Palavra "web": 728
  - ▣ Palavra "javascript": 688

# Trabalho em Equipa (não é)



**Realizar uma tarefa com  $n$  cabeças e  $2*n$  mãos**

- Como se garante que todos têm a mesma visão?
- Resultado podem ser  $n$  módulos não interoperáveis
- E se uma sub-tarefa afinal for demasiado extensa?

# Trabalho em Equipa (não é)



## **Dividir tarefas e compor resultado final**

- Resultados semelhantes ao anterior
- E se alguém não realiza trabalho (útil)?
- Só funciona em casos específicos.
  - ▣ Alta responsabilidade, baixa necessidade de integração

# Trabalho em Equipa (não é)



**Passar uma tarefa por várias mãos até estar feito**

- ❑ Aproximação tipo linha de montagem
- ❑ Criatividade muito limitada
- ❑ Um membro da equipa pode atrasar todo o processo
- ❑ Potencial fraca alocação de recursos: 1 trabalha, n-1 esperam.

# Trabalho em Equipa (não é)



**Criar um grupo e esperar que se faça trabalho**

- Indivíduos não trabalham só porque pertencem a algo!
- É preciso iniciativa e motivação intrínseca!



# Trabalho em Equipa (É)



- Trabalhar de forma cooperativa
  - ▣ Ajudar, não ser conflituoso
  
- Contribuir com esforço, ideias, sugestões
  - ▣ Não ser passivo ou negativo
  
- Comunicar!
  - ▣ Manter contacto com a equipa.
  - ▣ Errado: programador maravilha solitário.

# Trabalho em Equipa (É)



- Ter responsabilidade
  - ▣ Cumprir com o acordado pela equipa
  - ▣ Assumir erros
  
- Ter respeito pelos outros
  - ▣ Não atirar culpas
  - ▣ Não “minar” trabalho
  - ▣ Considerar posição dos colegas
  
- Tomar decisões
  - ▣ Equipa não funciona sem um líder!

# Trabalho em Equipe



- Muitos aspetos são chamados de “Soft Skills”
  - ▣ Atitude pessoal perante situações e eventos
  - ▣ Não tem a ver com inteligência ou conhecimento
  
- Aparece indiretamente no CV
  - ▣ Experiência passada em ambientes de trabalho em equipa

# Lei de Conway



A estrutura de uma aplicação de software  
reflete a estrutura da equipa que a construiu

- Melvin Conway

Corolário aplicado ao ensino:

A qualidade de um trabalho não depende só do  
conhecimento dos alunos, mas principalmente de como  
se organizaram.

Docente anónimo

# Lei de Brook



Adicionar mais pessoas a um projeto atrasado apenas o fará ficar ainda mais atrasado.

- Particularmente relevante para projetos de software
  - ▣ Exigida uma grande colaboração entre programadores
    - Ex: Linux, vários milhões de linhas de código para gerir
- São necessárias ferramentas e características individuais para se trabalhar bem em equipa!

# Papel da Tecnologia



- Tecnologia pode auxiliar funcionamento das equipas
  - ▣ Melhorar o “produto” final
  - ▣ Reduzir custo (esforço)
  
- Facilitar Planeamento:
  - ▣ Folhas de cálculo
  - ▣ Aplicações de planeamento de projeto
  - ▣ Existência de relatórios de outros projetos

# Papel da Tecnologia



- Facilitar comunicação
  - Mailing-lists
  - Grupos e fóruns online
  - Ferramentas para Video/Audio Conferência
  - Mito: têm de existir reuniões presenciais
  
- Facilitar colaboração Interna:
  - Repositórios de ficheiros
  - Identificação de tarefas pendentes
  - Documentação das ações

# Projetos de Software

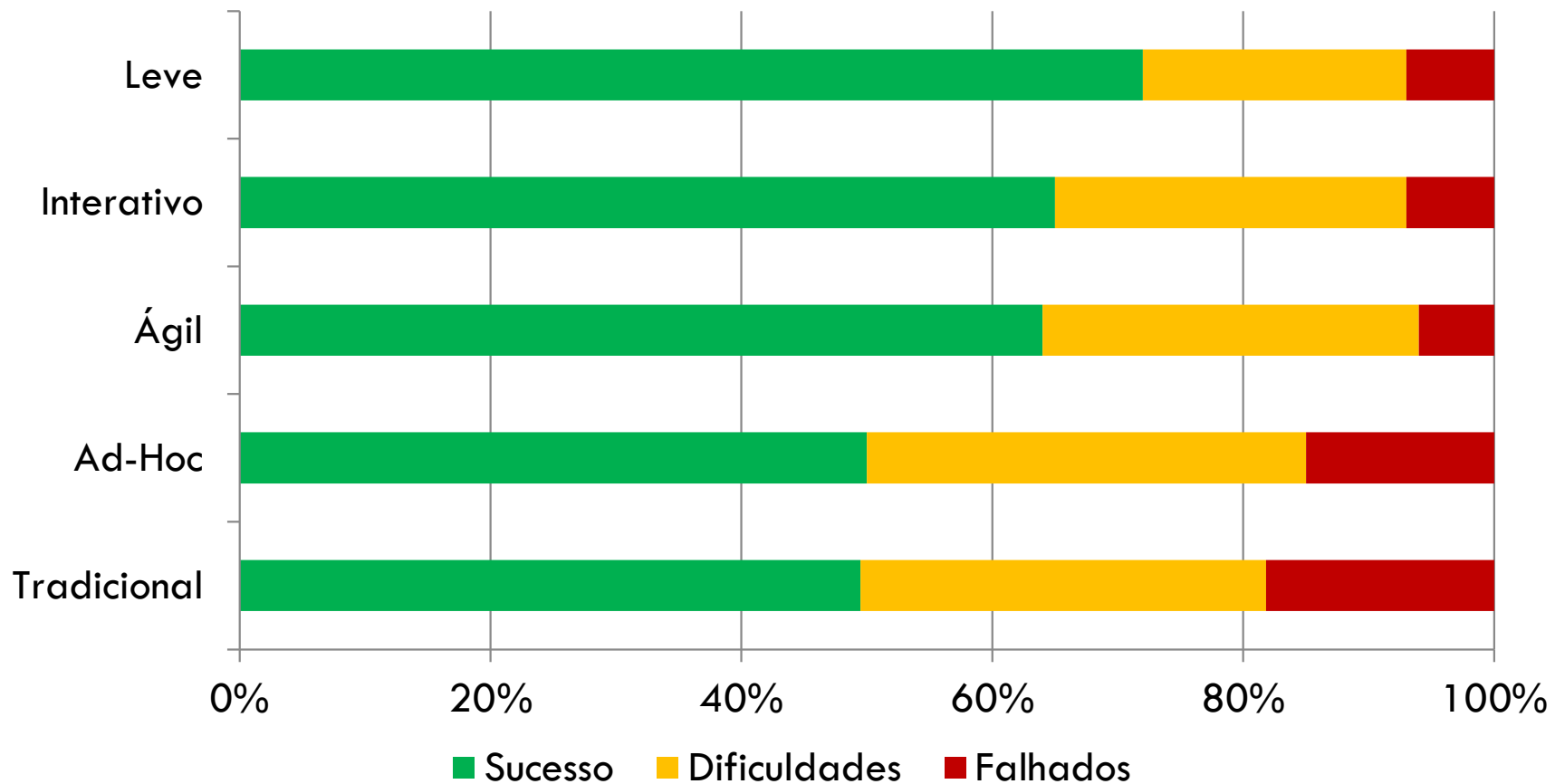


- ❑ 18% dos projetos são cancelados antes da sua conclusão
- ❑ 53% dos projetos custam mais ou demoram mais tempo
- ❑ 29% consideram-se como sendo bem sucedidos

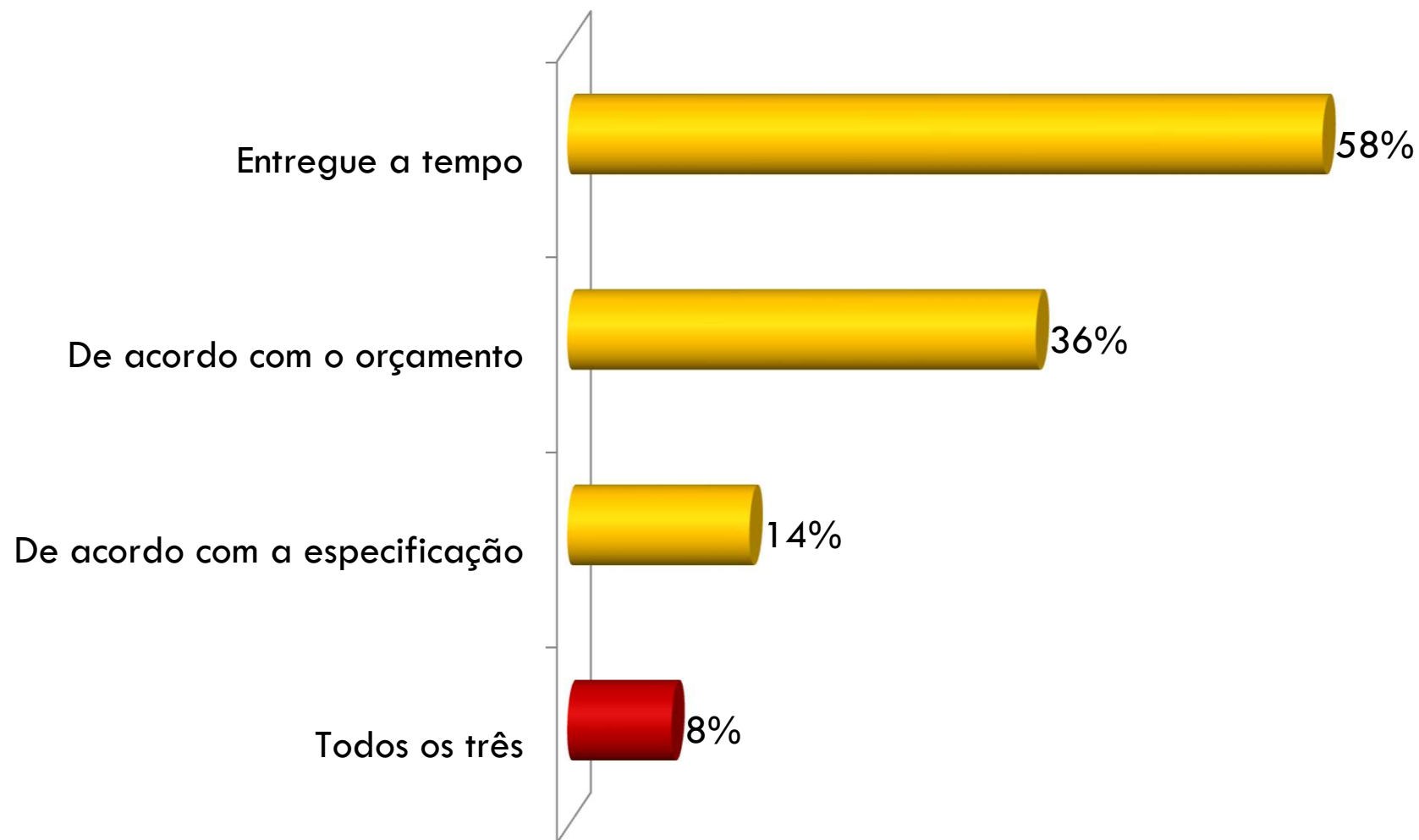


# Projetos de Software

## Sucesso dos projetos face à metodologia



# Projetos de Software



# Projetos de Software

- ❑ Sistemas extremamente complexos!
  - ▣ Necessário um controlo bastante grande sobre todos os processos.



15k-20k tijolos

Um tijolo pode fazer a casa ruir (?). Não

VS



Milhares a Milhões de linhas

Um carácter errado pode fazer o programa agir de forma não previsível

# Projetos de Software com falhas catastróficas



- NASA Mariner 1 (\$135 M)
  - ▣ Um carácter '␣' em falta no código
- Mars Climate Orbiter (\$1000M)
  - ▣ Controlo de propulsão em Libras, mas entendido em Newtons
- Knight Capital (\$10M/minuto durante 45 minutos)
  - ▣ Código pouco testado colocado em produção
- Therac-25 (Radioterapia) (5 mortes)
  - ▣ bug na definição da dosagem de radiação

# Projetos de Software



- Planeamento do trabalho autónomo
  - ▣ Prédefinição do modelo de interação
  
- Desenvolvimento independente
  - ▣ Testes autónomos e automáticos (unitários/funcionais)
  
- Integração
  - ▣ Integração de diferentes módulos
  
- Gestão de alterações (versões)

# Projetos de Software



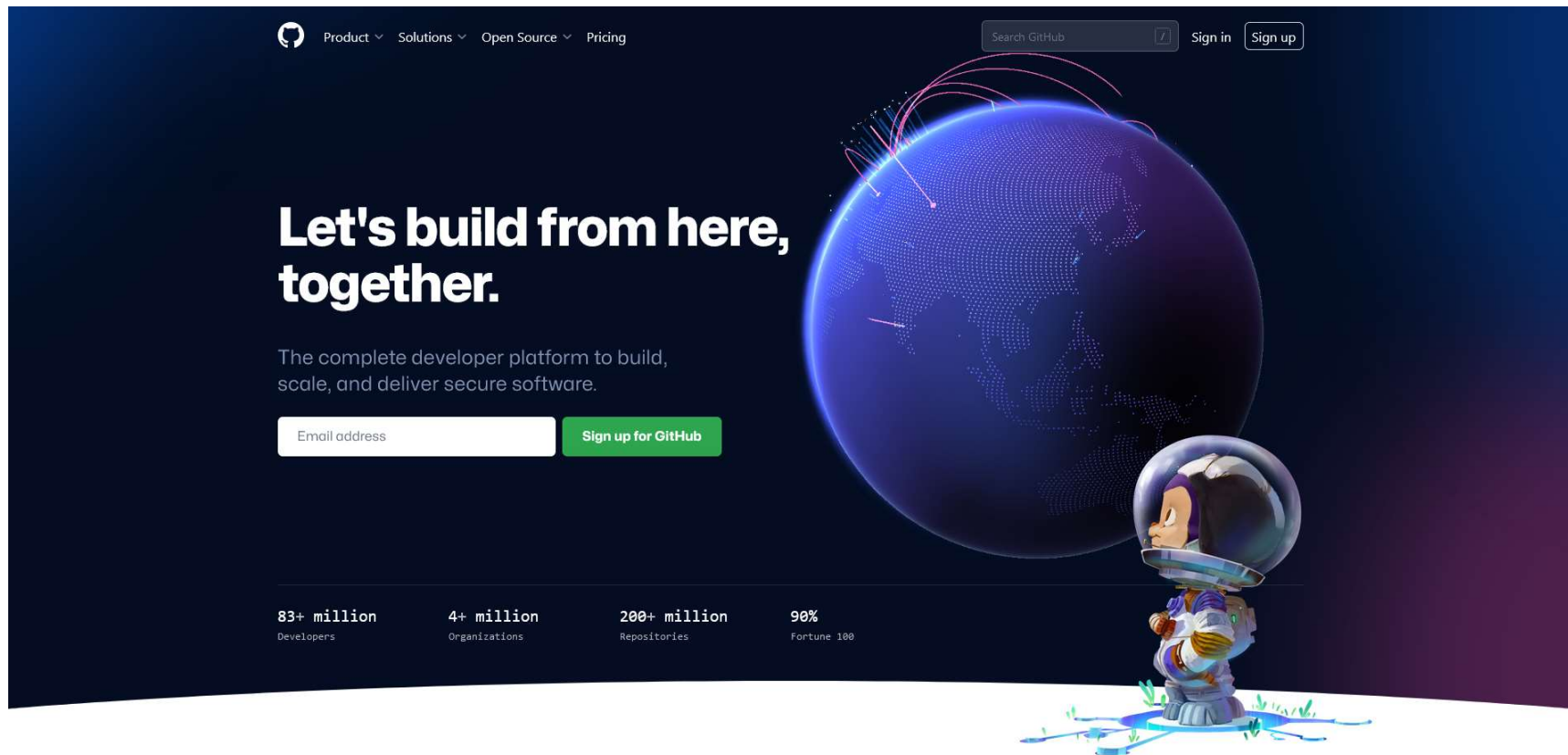
- Várias metodologias
  - ▣ Agile/Scrum, Waterfall, Spiral, OOP, RAD, etc...
  
- Foco na identificação das tarefas
  - ▣ Completa (Waterfall)
  - ▣ Cíclica: Agile, Spiral
  
- Foco em sistemas que permitam gerir todas as alterações ao código
  - ▣ Eventualmente: suportar revisão por terceiros

# Ferramentas



- hg, git, cvs, svn
  - ▣ Úteis para gerir fundamentalmente programas
  - ▣ Mas também servem para gerir alguns tipos de documentos
  
- Google Docs, Microsoft Word, LaTeX
  - ▣ Edição cooperativa de documentos
  - ▣ LaTeX: uso de “\input”
  
- skype, colibri, google hangout
  - ▣ Áudio/vídeo conferência
  
- Sistemas de messaging
  - ▣ Troca rápida de perguntas/respostas

# Ferramentas - GitHub





# Regras de Estilos



- Projetos têm de definir o estilo de programação
  - ▣ Todos os programadores devem aderir ao estilo
  
- Linguagens definem estilos próprios
  
- Âmbito
  - ▣ Nome dos ficheiros, métodos e variáveis
  - ▣ Espaçamentos
  - ▣ Indentação

# Para Referência



- Plataforma GitHub

- ▣ <https://github.com>

- Bugs de Software

- ▣ [http://en.wikipedia.org/wiki/List\\_of\\_software\\_bugs](http://en.wikipedia.org/wiki/List_of_software_bugs)