

# Gestão de projeto / produtos informáticos

---

Unidade 3 - Planeamento agile de produtos de software

[cedric.grueau@estsetubal.ips.pt](mailto:cedric.grueau@estsetubal.ips.pt)

# Objetivos de aprendizagem

---

- Compreender a importância do planeamento.
- Explicar o papel da incerteza na afetação de planos e estimativas de tarefas.
- Dividir tarefas em tamanhos apropriados e criar uma estrutura de divisão de produto (WBS) para um projeto.
- Conhecer as diferenças entre estimativas, metas e compromissos.
- Usar diferentes técnicas de estimativa, incluindo pontos de história, estimativas de velocidade e planeamento, time boxing e gráficos de Gantt.
- Gerar estimativas de trabalho para produtos de software usando uma variedade de abordagens, incluindo bottom-up, analogia, peritos e a fórmula de estimativa.
- Compreender as dependências entre tarefas, os diferentes tipos de dependências de tarefas e como elas afetam o planeamento do projeto.
- Criar gráficos CPM (Critical Path Method) e gráficos da técnica de revisão de avaliação de programas (PERT).
- Criar planos eficazes para o desenvolvimento de software, tanto aos níveis da iteração como de release.
- Identificar anti-padrões e tipos de riscos que podem afetar projetos.
- Avaliar e planejar os riscos do projeto.

# Planeamento agile de produtos de software

---

Introdução ao planeamento

# Objetivos de aprendizagem

## Introdução ao planeamento

---

- Explicar a importância do planeamento de um projeto.
- Descrever as palavras-chave na gestão de Produtos de Software: tarefa, função, cronograma, marco, produto de trabalho.
- Reconhecer e explicar um diagrama do Espaço de Incerteza (AUSP), e os conceitos que ilustra: significado da incerteza dos meios e da incerteza terminal, e como elas podem evoluir ao longo de um projeto.
- Definir uma estrutura de divisão do trabalho.
- Reconhecer como as etapas de início de projeto, aquisição de hardware / equipamento, desenvolvimento, teste e implementação podem ser representadas numa estrutura de divisão de trabalho.
- Criar uma estrutura de divisão de trabalho com base num exemplo.
- Identificar o tamanho adequado para uma tarefa.
- Explicar os termos: estimativa, meta e compromisso.
- Fornecer exemplos de estimativas, metas e compromissos.
- Compreender o que implica uma boa estimativa.

# Pergunta

---

Qual é para si a definição de um **tarefa**?

- A. Uma obrigação que uma pessoa assume ou desempenha
- B. Uma pequena etapa fácil de gerir de um projeto
- C. O resultado de um processo de software
- D. Um ponto de referência interno que mede o progresso

# Pergunta

---

Qual é para si a definição de um **tarefa**?

- A. Uma obrigação que uma pessoa assume ou desempenha
- B. Uma pequena etapa fácil de gerir de um projeto
- C. O resultado de um processo de software
- D. Um ponto de referência interno que mede o progresso

# Conceitos iniciais - Definições importantes

Palavra	Definição
<b>Tarefa</b> (Task)	Uma tarefa é uma etapa pequena e manejável de um projeto que deve ser concluída. As tarefas são essenciais para um projeto - tudo num projeto pode ser dividido em tarefas.
<b>Papel</b> (Role)	Um papel é um dever que uma pessoa assume ou desempenha em relação com o produto, Exemplos de funções incluem programador, operador de testes, designer, cliente e gestor de produto.
<b>Produto de trabalho</b> (Work Product)	Um produto de trabalho é uma saída/output produzido por uma tarefa ou processo,
<b>Agendamento</b> (Schedule)	Um agendamento é criado quando as tarefas de um projeto são mapeadas para uma linha de tempo.
<b>Marco</b> (Milestone)	Marcos referem-se a pontos de verificação internos utilizados para medir o progresso. Eles não são baseados no tempo, mas baseados em evento ou ação. Os marcos são mais usados em processos lineares e iterativos precoces de um projeto, Em ágil, os marcos não são geralmente usados. O progresso é, em vez disso, medido pelo software de trabalho em oposição a eventos ou ações. Releases e iterações tendem a serem baseados no tempo também, o que não se enquadra com marcos.

# Exemplo

---

Está a trabalhar com uma equipa de desenvolvimento que está a criar um website que permite que os estudantes possam colocar perguntas e receber ajuda de tutores em direto. Os alunos são também encaminhados para vídeos de tutoriais e ligações para referências. Está a agendar tarefas. Deve afetar a tarefa “**Escrever código fonte para conversas ao vivo com os tutores**”.

Qual das seguintes tarefas deve ser agendada antes dessa tarefa?

- A. Carregar vídeos de tutoriais para os tutores referenciarem
- B. Executar testes de aceitação para as conversas ao vivo com os tutores
- C. Desenhar a ferramenta de live chat
- D. Contratar tutores para trabalhar nas conversas ao vivo



# Exemplo

---

Está a trabalhar com uma equipa de desenvolvimento que está a criar um website que permite que os estudantes possam colocar perguntas e receber ajuda de tutores ao vivo. Os alunos são também encaminhados para vídeos de tutoriais e ligações de referências. Está a agendar tarefas. Deve afetar a tarefa **“Escrever código fonte para a ferramenta de conversas ao vivo com os tutores”**.

Qual das seguintes tarefas deve ser agendada antes dessa tarefa?

- A. Carregar vídeos de tutoriais para os tutores referenciarem
- B. Executar testes de aceitação para as conversas ao vivo com os tutores
- C. Desenhar a ferramenta de live chat
- D. Contratar tutores para trabalhar nas conversas ao vivo

# Agendamento

Schedule

---

- **Consiste no mapeamento de tarefas num linha temporal**
- Passo importante: Reconhecer as dependências entre tarefas
- Mapear as tarefas de forma a que os produtos de trabalho sejam prontos antes de serem precisos
- Garantir que os requisitos com prioridades altas sejam criados antes
- Planear o que estará concluído em cada iteração
- Complicado?
  - -> Existem técnicas para ajudar

# Marcos

## Milestones

---

- **Ponto de referência interno para medir o progresso**
- Não são baseados no tempo mas em ações ou eventos
- Mais comuns em processos lineares ou processos iterativos iniciais
- As práticas ágeis tendem a não ter marcos

# Exemplo

---

Joana está a trabalhar num projeto que permite aos utilizadores monitorizar e rastrear as despesas mensais. Joana e a equipa de desenvolvimento decidem medir o progresso da equipa com base em marcos.

Qual dos seguintes pontos é um marco?

- A. O fim de um sprint de 2 semanas
- B. O aniversário que celebra o primeiro ano do projeto
- C. A data de entrega prevista do projeto
- D. Os requisitos foram escritos e revistos

# Exemplo

---

Joana está a trabalhar num projeto que permite aos utilizadores monitorizar e rastrear as despesas mensais. Joana e a equipa de desenvolvimento decidem medir o progresso da equipa com base em marcos.

Qual dos seguintes pontos é um marco?

- A. O fim de um sprint de 2 semanas
- B. O aniversário que celebra o primeiro ano do projeto
- C. A data de entrega prevista do projeto
- D. Os requisitos foram escritos e revistos

# Como criar tarefas?

---

- As tarefas existem logo em fases iniciais do projeto
  - Criar requisitos
- Uma vez criados e ordenados por prioridades, os requisitos devem ser “partidos” em tarefas de desenvolvimento

# Exemplo

---

- Imagine que estamos a criar uma aplicação que permite aos utilizadores consultar versões digitais de livros, do tipo livraria virtual. Consideramos que um dos requisitos é o seguinte:

Como: um utilizador,

Quero: procurar com base em palavras-chave

De modo que: posso encontrar o livro que contem o texto que procuro

- Como transformar o requisito em tarefas?
  - Desenhar a ferramenta de pesquisa de palavras-chave
  - Adquirir ou desenvolver um algoritmo de pesquisa com base em palavras-chave

# Organização de tarefas

---

- Na disciplina de ESA viram que:
  - as tarefas compõem atividades
  - As atividades compõem fases
- Cada fase possui tarefas que são executadas
- As práticas ágeis fornecem formas de organizar as tarefas de maneira a que elas estejam efetivamente executadas
  - Scrum - plano de iterações
  - Kanban - forma visual de rastrear o progresso das tarefas através de um conjunto de etapas



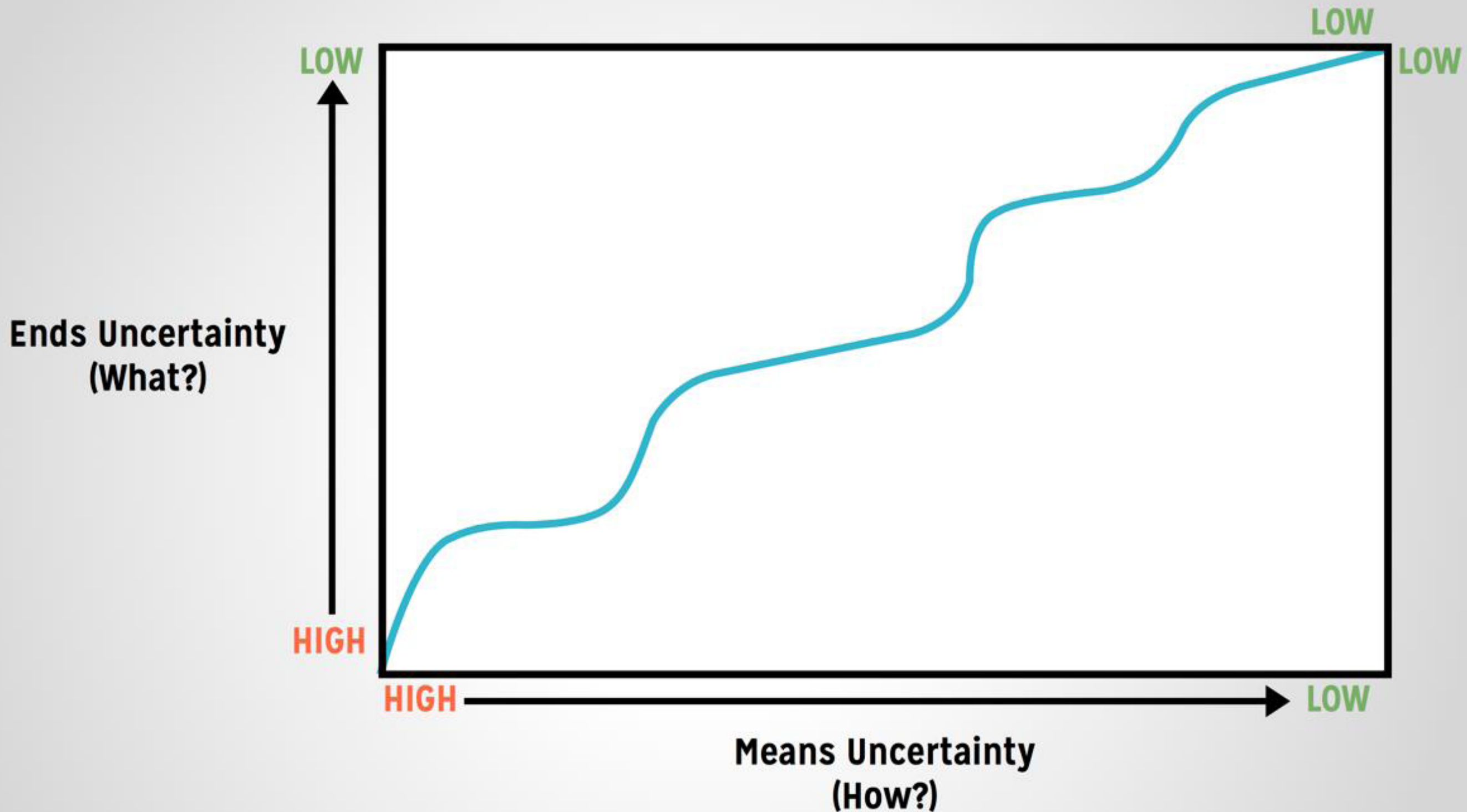
# O espaço de incerteza

---

- Incerteza
  - Onde começamos?
  - Onde terminamos?
  - Como a equipa vai desenvolver?
  - O que a equipa va desenvolver?
- 2 tipos de incerteza (Alexander LaFleur)
  - Incerteza dos meios
  - Incerteza final
- Diagrama do espaço da incerteza (Mike Cohn)

# O espaço de incerteza

---



# Pergunta

---

Em que localização deste diagrama gostaria que o seu produto se encontre no fim do seu projeto?

- A. Incerteza de meios elevada, incerteza final elevada
- B. Incerteza de meios elevada, incerteza final baixa
- C. Incerteza de meios baixa, incerteza final elevada
- D. Incerteza de meios baixa, incerteza final baixa

# Navegar ao longo do diagrama do espaço de incerteza

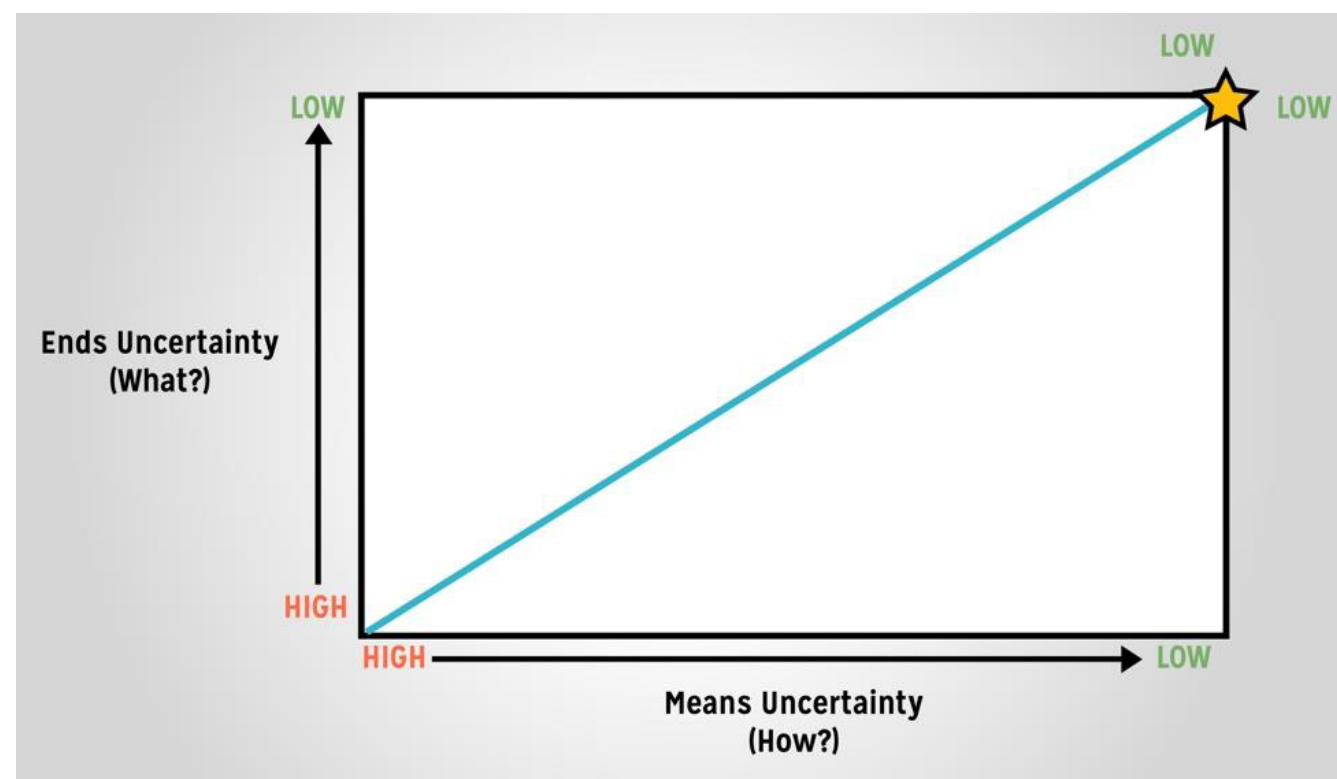
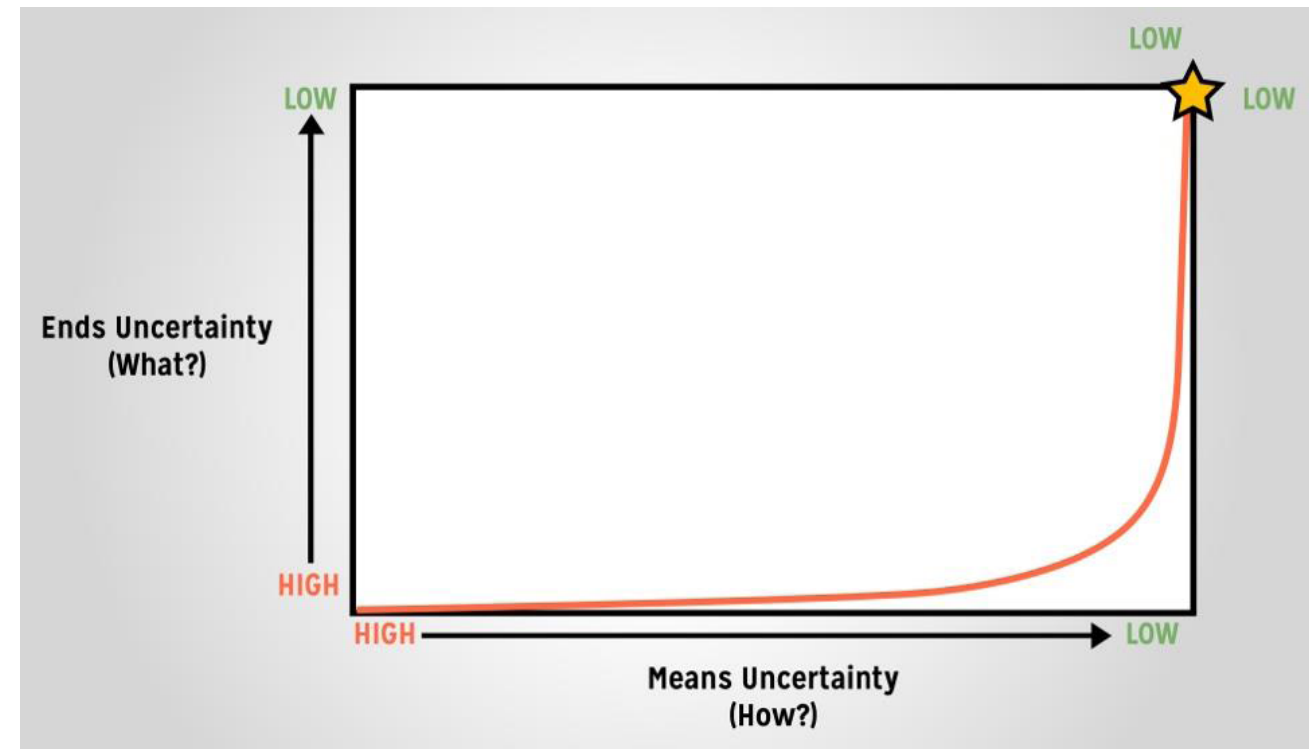
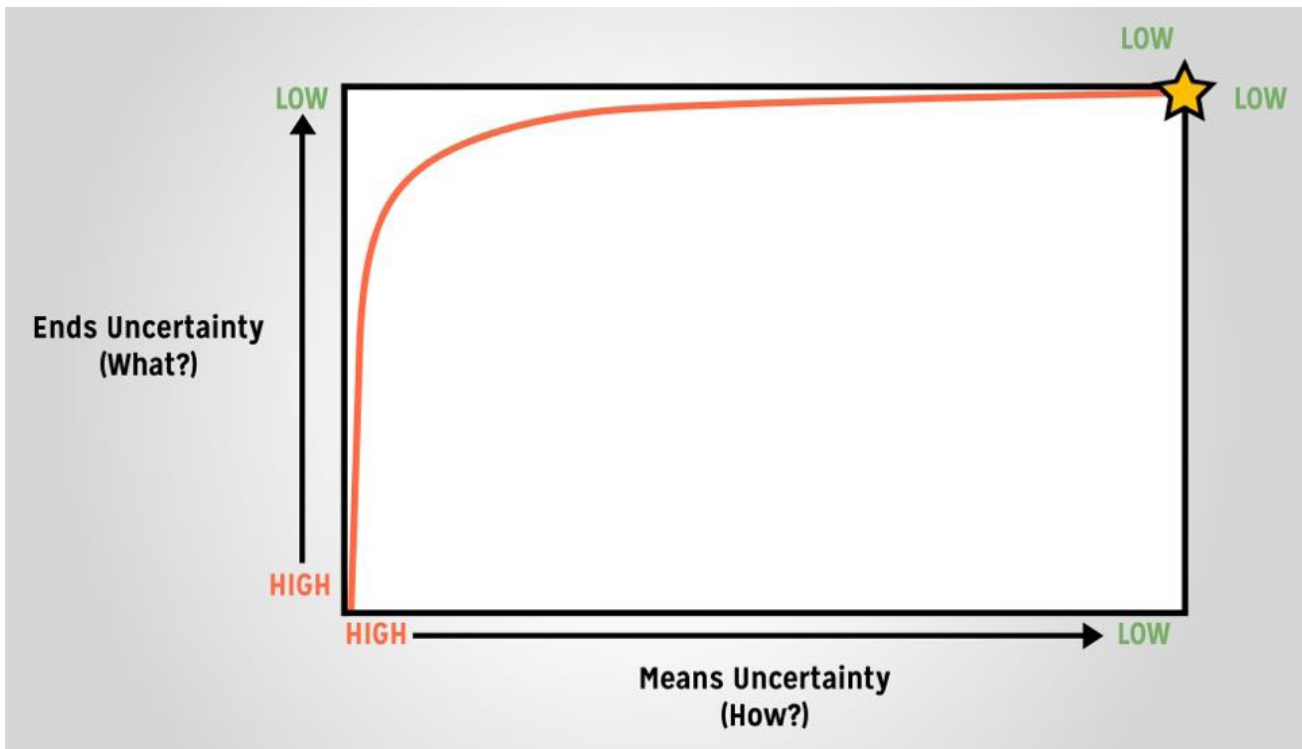
---

Há muitas maneiras diferentes de navegar através de projetos usando o diagrama de espaço de incerteza:

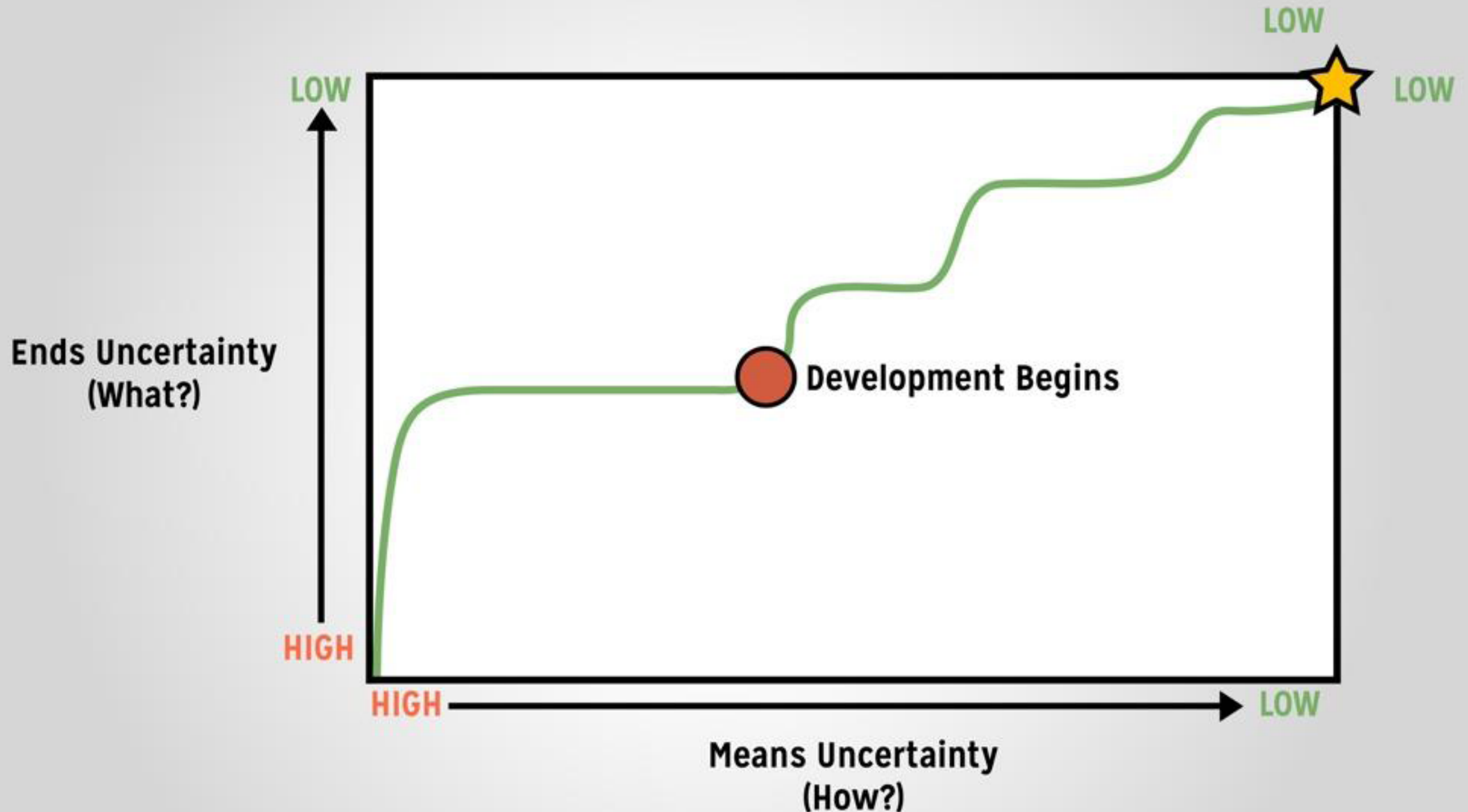
1. Determinar primeiro o **"o que"** de um projeto, depois o **"como"**.
  2. Determinar **"como"** desenvolver um projeto e depois criá-lo à medida que se desenvolve.
  3. Determinar tanto o **"o quê"** quanto o **"como"** de um projeto com o maior equilíbrio possível entre os dois.
- A primeira forma identifica o que o projeto irá criar e, de seguida, como criá-lo. Usando este método, os requisitos são todos determinados em primeiro lugar, e em grande detalhe, antes de quaisquer planos para completá-los sejam desenvolvidos.

# Navegar diagramas de espaço de incerteza

Qual dos diagramas correspondem às abordagens 1, 2 e 3?



# A navegação ágil no espaço de incerteza



# Nota a cerca dos espaços de incerteza

---

- Na maioria dos casos do mundo real, os clientes e a equipa de desenvolvimento definirão a maioria dos requisitos para o projeto antes do início do planeamento. Durante este processo, pode haver menos incerteza final do que incerteza de meios. No entanto, uma vez que o desenvolvimento do produto começa e a parte do planeamento do "como" aumenta, a incerteza dos meios diminuirá.
- À medida que os requisitos são adicionados ou alterados durante o curso do projeto, pode haver uma ligeira flutuação na direção da incerteza das extremidades inferiores, seguida de menor incerteza dos meios. O desenvolvimento e as soluções mudam à medida que os requisitos mudam, de modo que a incerteza se parecerá mais com o diagrama anterior, ao invés de uma linha reta.
- Ambas as incertezas final e de meios devem caminhar para se tornarem baixas. À medida que as mudanças ou os requisitos adicionais ocorrem e que qualquer mudança correspondente no desenvolvimento aconteça rapidamente, o projeto permanecerá adaptável e evitará desperdício de tempo e recursos.
- Não importa o método de navegação no espaço de incerteza, **as tarefas são necessárias para que o projeto alcance seu fim.**

# Desenvolvimento de produto

---

- Novo projeto : **assustador no início**
  - desenho, escrita de código, testes, ...
  - marketing, apoio técnico e gestão de sistemas, ...
- Solução:
  - dividir, planejar, gerir





# Etapas do planeamento

---

## Abordagem preditiva

1. calcular o prazo
2. calcular o custo
3. identificar as atividades
4. calcular a duração das atividades
5. ordenar as atividades
6. estabelecer o planeamento
7. ajustar o planeamento

## Abordagem ágil

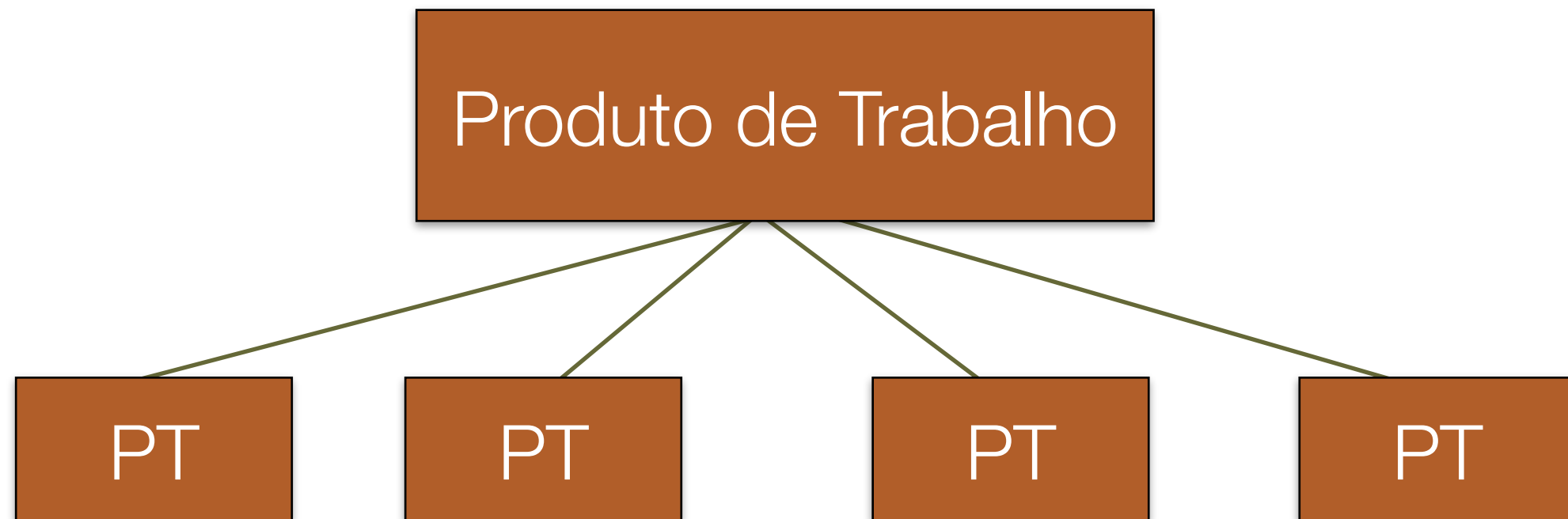
1. Estimativa relativas das funcionalidades'
2. Atribuição das funcionalidades às várias releases (roadmap)
3. (release) clarificação das funcionalidades (user stories) e afetação às várias iterações
4. (iteração) levantamento das tarefas para user story tratada
5. (daily stand-up meeting ) ajustamento do planeamento, atribuição das tarefas

# Estrutura de divisão do trabalho

Work Breakdown Structure (WBS)

---

- Dividir um projeto em tarefas organizadas em hierarquia



Dividir os PT até que possam ser atribuídos à uma pessoa

# Exemplo de estrutura de divisão do trabalho

---

- Imagine que está a dividir um projeto de construção de uma casa, especificando a tarefa de ligar o sistema eléctrico no interior da casa.
- É possível dividir cada tarefa em peças cada vez mais pequenas até um certo nível:
  - retirar 1,5 cm da ponta do cabo
  - inserir o cabo no receptáculo certo
  - ....
- Esta divisão deixa de fazer sentido quando se chega a especialidade do papel em causa.
- Nesta caso, o gestor (o construtor) pode se limitar a tarefa **instalar o sistema eléctrico**.

# Pergunta

---

Paulo é um gestor de produto que trabalha num jogo de vídeo animado. Está a trabalhar na estrutura de divisão do trabalho e não tem a certeza até que ponto deve dividir os seus produtos de trabalho. O jogo de vídeo pretende ter gráficos excelentes, incluindo a cena de background noturno que usa um céu repleto de estrelas.

**Quais dos seguintes produtos de trabalho teriam um tamanho adequado?**

- A. Um jogo de vídeo
- B. Os Desenhos do jogo
- C. Os backgrounds do jogo
- D. A cena de background noturna
- E. Uma estrela

# Exemplo

construção de uma aplicação de notícias

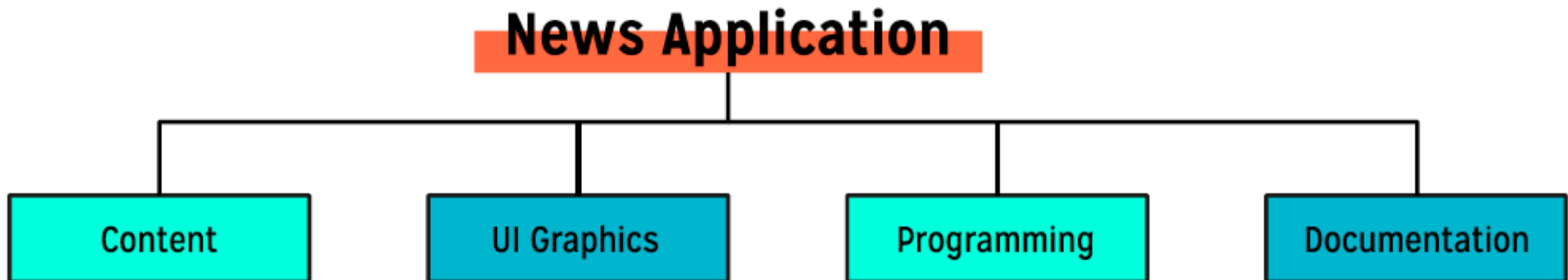
---

- Mostrar as notícias para uma empresa de notícias global
- Os utilizadores podem consultar as manchetes, ler os artigos e ver diferentes secções (finanças, lazer, ...)
- Widgets live: atualizações (ações, resultados desportivos, ...)
- Produto de trabalho: aplicação de notícias

# Exemplo

construção de uma aplicação de notícias

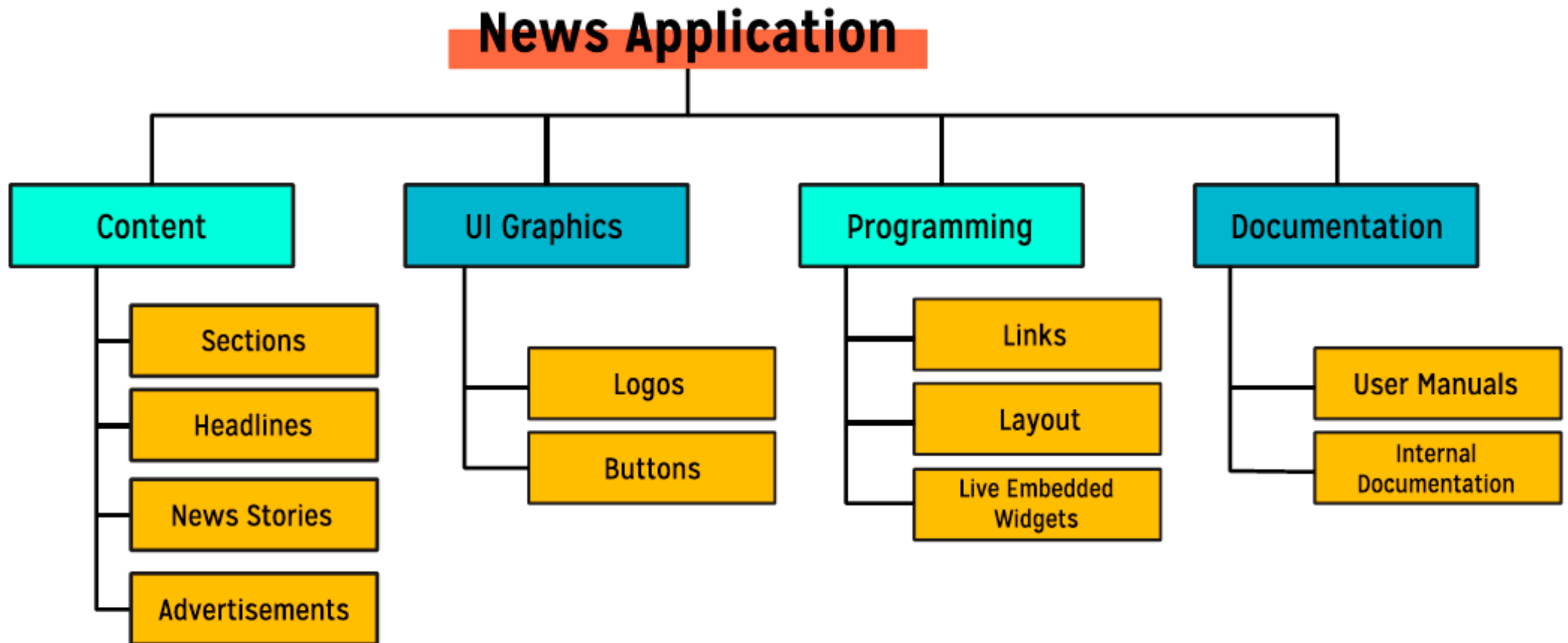
---



# Exemplo

construção de uma aplicação de notícias

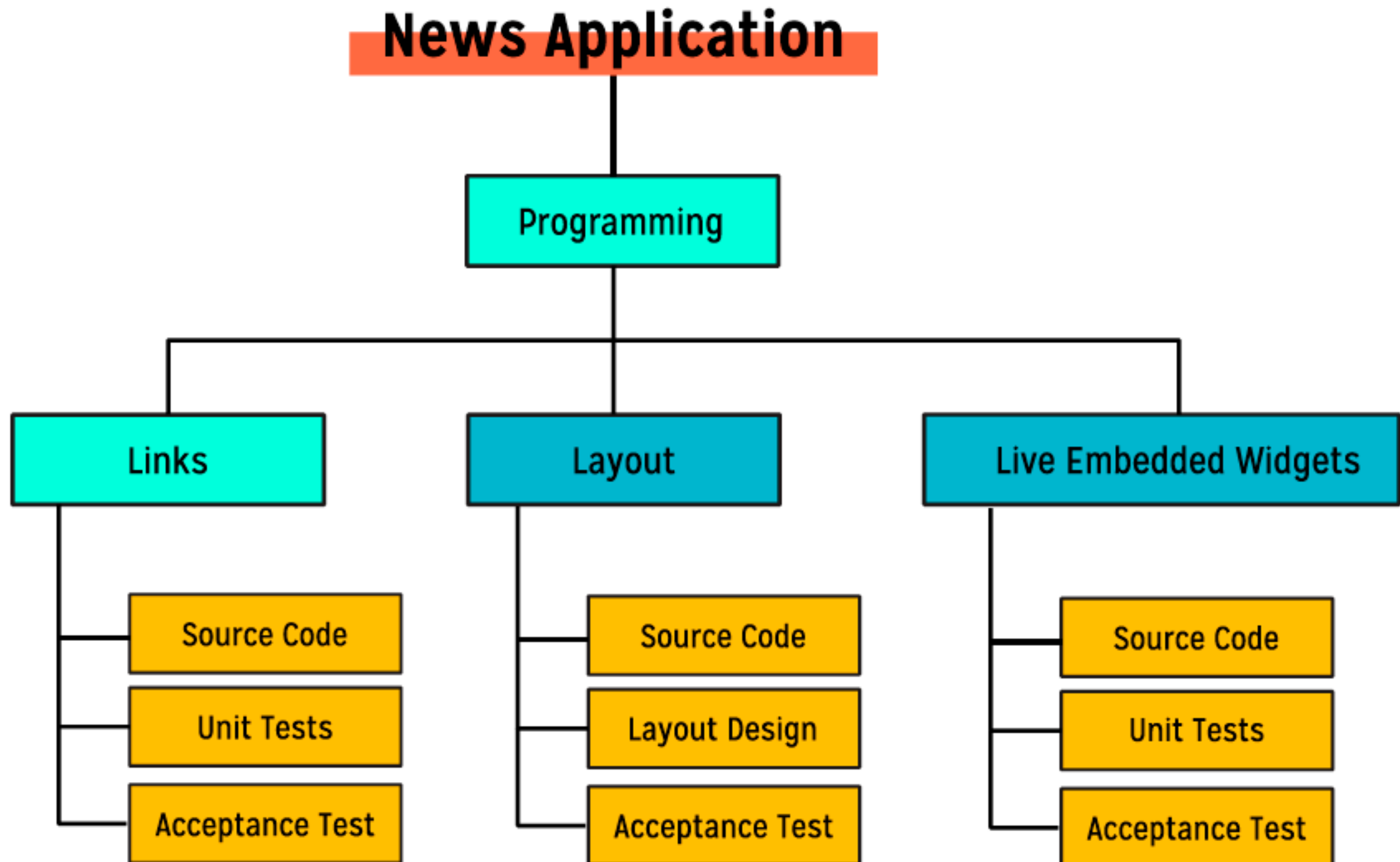
---



# Exemplo

construção de uma aplicação de notícias

---





# Exemplo

## construção de uma aplicação de notícias

---

- **Testes unitários** servem para testar a funcionalidade de baixo nível do código-fonte;
  - Eles devem ser escritos antes de código-fonte: **dependência entre tarefas**
  - Para o ramo "Links", um teste unitário poderia determinar se os links estão a funcionar de forma adequada, por exemplo.
- **Testes de aceitação**, por outro lado, permitem verificar a aplicação como um todo.
  - Neste mesmo sub-ramo, um teste de aceitação poderia testar a interface para se certificar de que todos os links são funcionais e corretos, por exemplo.
- Desenhar o layout é outro produto de trabalho que deve ser listado, bem como todos os testes de aceitação associados ao projeto que garantem que o layout é compatível e funcional em vários dispositivos.
- Estas pequenas tarefas de criação de desenhos e de escrita de testes são produtos de trabalho de tamanho adequado. Eles são muito mais fáceis de gerir do que uma grande tarefa como "criar uma aplicação de notícias" seria.
- Eles também estão no âmbito, ao contrário de uma tarefa como "criar todo o código fonte para sala de chat do leitor."
- Uma tarefa como a última não deve ser incluída numa estrutura de divisão de trabalho, porque é muito grande para abordar facilmente e pode facilmente resultar na perda de tarefas menores.

# Pergunta

---

Podemos criar as tarefas com base nos Produtos de trabalho definidos para a aplicação

**Quais das seguintes tarefas estaria associada com o exemplo anterior?**

- A. Escrever testes unitários para os widgets live encapsulados
- B. Criar uma aplicação de notícias
- C. Criar o desenho para o layout da interface gráfica
- D. Adicionar conteúdos a aplicação
- E. Criar o código fonte para a sala de chat do leitor

# Criar uma estrutura de divisão do trabalho

---

- Projetos de grande dimensão podem ter centenas ou milhares de tarefas
- Usar um gráfico em árvore WBS simples semelhante ao nosso exemplo pode rapidamente tornar-se pesado.
- Muitas outras ferramentas estão disponíveis para ajudar a criar estruturas de divisão de trabalho em escalas maiores, incluindo Microsoft Excel e Microsoft Project. Ambos são mais adequados para projetos de grande escala.
- Estruturas de divisão de trabalho também não se limitam a preocupações de gestão de produtos em torno da fase de Design e Implementação.
- Podem também incluir atividades de fase de Especificação, atividades de fase de Verificação e Validação, promoção marketing do produto, organização de suporte técnico, análise de dados, criação de servidores, etc.

# Utilizações de estruturas de divisão do trabalho

---

- Determinar as tarefas com base na EDT (produtos de trabalho pequenos e fáceis de gerir -> tarefas pequenas e fáceis de gerir)
  - Produto de trabalho “Testes unitários” do produto de trabalho “Links”-> tarefa “Escrever testes unitários para os links”
- Determinar os riscos potenciais associado a um determinado produto de trabalho: ajuda a listar os riscos
- Demonstrar o produto ao cliente: visão geral do produto com as *features* de cada produto de trabalho
- Ajudar a determinar a data de conclusão do projeto: mais fácil de determinar a duração de tarefas pequenas

# Outras utilizações de estruturas de divisão do trabalho

---

- Apoio ao orçamento
  - Identificar os recursos necessários para as várias atividades e tarefas
  - Identificar o esforço por recurso
  - Identificar os custos associados aos recursos
- Top-Down
  - Determinar o esforço e a duração (quantos dias, quantas semanas)
- Bottom-up
  - Determinar as tarefas chave

# Estimativas, Metas e Compromissos

---

- “Estimo que podemos comprometer-nos com uma meta de data”



- Palavras usadas de forma inadequada no dia a dia mesmo entre membros de equipes de desenvolvimento.

# Estimativas

---

- Uma **aproximação do tempo** que levará a equipa de desenvolvimento ou o programador para completar uma tarefa.
- Uma boa estimativa é geralmente baseada numa tarefa bem-formada
- Baseada em dados como por exemplo tempos anteriores ou tarefas semelhantes
- Deve ser um intervalo (um valor preciso em horas não poderá ser preciso)
- **Não negociável**

# Exemplo

---

- Miguel é um gestor de produto, inquire um dos seus programador, Luisa, para saber quanto tempo irá demorar para desenvolver uma certa feature.
- Miguel já trabalhou com a Luisa e sabe que se trabalhar arduamente irá demorar 3 horas
- Luisa, pensa e decide que irá usar uma almofada para não ter de trabalhar arduamente e responde ao Miguel que irá precisar de 9 horas.
- Miguel responde que acha que irá demorar apenas 3 horas e pede para dividir a diferença.
- A Luisa aceita e Miguel regista 6 horas.

**entre 3 e 9 horas**



# Pergunta

---

É um gestor de produto que trabalho com uma equipa pequena de programadores. Decide consultar os programadores para determinar um tempo de desenvolvimento para um novo produto. Vai falar com o programador Rui e pergunta quanto tempo irá demorar para escrever testes para uma determinada feature. Estima que vai demorar 5 horas já que o Rui acaba de terminar uns testes que escreveu para uma feature semelhante e que demorou 6 horas. O Rui quer se dar alguma segurança e responde-lhe 10 horas.

**O que seria uma estimativa precisa para esta tarefa?**

- A. 5 horas
- B. 6 horas
- C. 10 horas
- D. 7,5 horas

# Pergunta

---

Você é um gestor de produto que trabalha com uma equipa pequena de programadores. Decide consultar os programadores para determinar um tempo de desenvolvimento para um novo produto. Vai falar com o programador Rui e pergunta quanto tempo irá demorar para escrever testes para uma determinada funcionalidade. Estima que vai demorar 5 horas já que o Rui acaba de terminar uns testes que escreveu para uma funcionalidade semelhante e que demorou 6 horas. O Rui quer se dar alguma segurança e responde-lhe 10 horas.

**O que seria uma estimativa precisa para esta tarefa?**

- A. 5 horas
- B. 6 horas
- C. 10 horas
- D. 7,5 horas

# Estimativas

---

- Abordagens para criar estimativas
  - Bottom-up
  - Analogia
  - Peritos
  - Usar uma formula de estimativa
- Problemas que podem afetar a capacidade de fazer uma boa estimativa incluem:
  - Falta de experiência do programador com tarefas semelhantes
  - Demasiada atenção aos detalhes nas tarefas que levam a sobre-estimativas
  - Não há informações suficientes disponíveis sobre a tarefa para criar uma estimativa precisa

# Metas

---

- Uma meta é um ponto específico no cronograma para o projeto, ou parte do projeto, que se pretende alcançar.
- É quase um prazo ideal, e é geralmente definido externamente para a equipe de desenvolvimento.
- Tal como as estimativas, os objectivos não são negociados. Uma vez que um alvo foi decidido, ele deve ser alcançado para o sucesso do produto e satisfação do cliente.
- Na verdade, as metas são muitas vezes contratuais e podem ser usadas na organização de promoções dos produtos.
- Exemplos de metas
  - Fim de um sprint
  - Data de lançamento de um produto.

# Compromissos

---

- Um compromisso é uma coisa que concorda em entregar.
- Com base em estimativas e metas, um compromisso pode ser negociado.
- Uma boa pergunta a fazer quando se determinam compromissos é:

"Como me posso comprometer no que a equipa de desenvolvimento se compromete a fazer, com base nas suas estimativas e as restrições de tempo (do projeto)?"

# Exemplo de utilização de estimativas, metas e compromissos

---

- Um projeto teórico pode ser estimado pela equipa de desenvolvimento como necessitando de 1.000 horas de trabalho, com base nos requisitos que o cliente expressou. Esta é uma **estimativa**.
- O cliente, entretanto, pode requerer que o projeto termine dentro de 600 horas. Este é uma **meta**.
- Após a negociação, o cliente e a equipa de desenvolvimento podem ajustar o âmbito do projeto de forma a que passa a incluir cerca de 500 horas de trabalho. Este é um **compromisso**.
- O projeto pode levar mais ou menos tempo para concluir do que o compromisso, e o tempo real pode ser usado como dados para estimativas futuras.

## Boas práticas:

---

1. As estimativas devem ser discutidas com a equipa de desenvolvimento em primeiro lugar, com valores tão precisos quanto possível.
2. Deve haver discussões com o cliente para determinar as metas.
3. Com base nessas discussões, os compromissos podem ser determinados tanto com a equipa de desenvolvimento como com o cliente.

# Notas sobre estimativas, metas e compromissos

---

- Ambas as estimativas e metas não são negociáveis uma vez definidas,
- É comum que as estimativas sejam convertidas automaticamente em compromissos e metas, mas isso é uma má prática. Se um projeto é estimado para levar 1.000 horas de trabalho, muitas vezes os clientes automaticamente assumem que este é um compromisso de 1.000 horas. Isso muitas vezes leva à tentação de inflar as estimativas, também uma má prática.
- É importante que as estimativas, metas e compromissos sejam claros e separados para o cliente e para a equipa de desenvolvimento durante as suas discussões. Se isso puder ser feito, é mais provável que o projeto e suas metas possam ser executados conforme planeado, sem que o âmbito se torne muito grande, pois garante que as metas possam ser alcançadas. Considerar que 1.000 horas é uma estimativa, e a meta é de 600 horas ajudou a criar um compromisso de 500 horas. Este compromisso ajudou a reduzir o escopo do projeto para ficar dentro da meta.



# Caso prático

---

- A empresa X, especializada na publicação de informações de referência, enfrenta novos desafios para os próximos anos: publicação de publicações mais frequentes e mais reactivas, fontes de informação mais diversificadas e em quantidades crescentes, aumento de riqueza de informações publicadas para enfrentar a competição.
- O sistema de informação anterior não abordou essas necessidades:
  - Composto por aplicações heterogêneas esparsas, obrigou os utilizadores a fazer malabarismos de uma aplicação para outra, tornando a realização das suas tarefas ainda mais laboriosas.
  - As atualizações foram difíceis de monitorizar devido a pouca visibilidade nas diferentes atualizações consolidadas por diferentes ferramentas independentes.
  - A publicação dos livros foi difícil de perceber porque exigia um "congelamento" durante o qual as atualizações estavam suspensas.

# Caso prático

---

- Para superar os limites deste sistema, a sociedade X desenvolveu um novo sistema caracterizado por:
  - A implementação de um fluxo de trabalho: os usuários não precisam mais procurar as atualizações que devem fazer; cada informação fonte alimenta a “todo list” dos atores envolvidos. A informação está agora "empurrada" para o utilizador.
  - Um "escritório de negócios": aplicações anteriormente heterogêneas tornam-se componentes do mesmo ambiente e estão integradas num escritório de negócios; eles partilham uma interface e funcionalidades comuns e a capacidade de se comunicar uns com os outros.
  - Uma gestão das versões de informações de referência: permite a dissociação completa entre as operações de atualização e de publicação de livros, evitando o fenómeno de "congelamento".

Requisitos						
id	Requisito	tipo	prioridade	Risco	Custo	Estado
F01	o sistema deve permitir a autenticação de usuários	F	1	1	1	done
F02	o sistema deve permitir a criação de novos usuários	F	3	3	2	todo
F03	o sistema deve permitir a recuperação de senhas	F	2	3	3	todo
F04	o sistema deve permitir a exclusão de usuários	F	1	3	3	done
F05	o sistema deve permitir a alteração de senhas	F	1	2	3	done
F06	o sistema deve permitir a recuperação de e-mails	F	2	3	2	doing
F07	o sistema deve permitir a alteração de e-mails	F	1	3	1	done
NF01	o sistema deve ser seguro	NF	3	1	3	todo
NF02	o sistema deve ser escalável	NF	2	2	2	todo

# Product Backlog 98

---