# Etapas de desenvolvimento – Estimativas

Imagine que estamos elaborando uma fase do jogo e vamos criar um protótipo para ela. Temos os requisitos e os recursos listados e precisamos planejar a entrega. Como fazemos para completar o cronograma de atividades alocando os recursos e os prazos para as entregas?

Você deve estar pensando que não tem condições de estimar o tempo de desenvolvimento de nenhuma tarefa, pois ainda tem pouca experiência. Existem técnicas e processos que permitirão que você estime, junto com o produtor e os líderes, o tempo necessário para desenvolvê-las.

Esse processo é feito inicialmente pelo produtor, junto com a equipe de *marketing*, a partir da definição da entrega final e das datas das etapas de desenvolvimento. Com a estimativa inicial dada pelo gerente de projeto, o líder da equipe deve avaliar, junto com todos, os prazos estabelecidos. Se alguém não souber estimar, o membro mais experiente deve fazê-lo. Mas se ele estimar e você não atender à estimativa? Se isso ocorrer no início do projeto, não haverá problema, pois, a partir desse momento, você tem um parâmetro de sua velocidade.

Chandler (2012) enfatiza a técnica de time-boxing, que consiste em estabelecer um tempo fixo para a execução das tarefas. A tarefa deve estar bem definida e clara, com os requisitos estabelecidos, com o período de início e término definidos. Então o líder deve verificar o quanto da tarefa foi concluído no tempo estabelecido. A partir daí, passa-se a ter um parâmetro-base, a partir do qual se evolui a técnica. Em Novak (2008), Jesper Sorensen, um experiente produtor de games, argumenta que oportunizar aos desenvolvedores a realização da estimativa de tempo do seu trabalho os mantêm mais motivados do que o se líder estimasse para eles.

Chandler (2012) aconselha, ainda, não programar horas extras, mas considerar férias, licenças em função de doenças e feriados, além de contemplar de 5 a 6 horas diárias de trabalho, em função de reuniões, verificação de *e-mails* e serviços não

relacionados às tarefas.

Vamos olhar, agora, para a estimativa utilizando técnicas de metodologias ágeis. Lembrando que com a abordagem de *Scrum* temos as histórias, que consistem no que deve ser feito. Para realizar a estimativa dessas histórias, é verificado o tamanho delas (o tempo necessário) em termos de desafios que apresentam, considerando toda a equipe envolvida na sua realização, ou seja, recursos das áreas de *design*, arte, programação e QA. A ideia de envolver toda a equipe tem como objetivo que as dúvidas relativas à sua implementação possam ser tiradas de forma mais rápida.

Mas quanto tempo devemos gastar estimando o tamanho de uma história? Quanto mais tempo se investe estimando, mais apurado é o resultado. Mas isso tem um limite, pois, a partir de um certo ponto, em termos de esforço, começamos a perder resultado. Para estimar as histórias, devemos quebrá-las em histórias menores, mas não devemos, por exemplo, focar-nos em detalhes de implementação, como quantas variáveis vamos utilizar. Este nível de detalhe varia ao longo da codificação, consistindo em um esforço desperdiçado.

Adicionalmente, utiliza-se a comparação com as estimativas feitas e consolidadas de histórias semelhantes, porém uma maior e outra menor. Se estamos trabalhando em um jogo de FPS e vamos programar a utilização de uma arma, comparamos com a história referente à utilização de outra arma já elaborada. Atente-se para sempre comparar com uma história maior e outra menor para obter uma média entre elas e atribuí-la à história que implementará.

Quando trabalhamos usando *Scrum*, temos a técnica de estimativa do tamanho de histórias indicando a sua complexidade. Mas não se trata de estimar o tempo necessário para completá-la. Apesar de neste processo o fator tempo manter-se sempre em mente, ele é definido ao dividir a história em mais detalhes, visando à sua execução no planejamento da sprint (sprint planning).

Lembre-se de que uma sprint pode durar duas semanas ou um mês e, assim, devemos estimar as nossas histórias para serem executadas dentro desse prazo. Elencamos as prioritárias em função do retorno do valor para o cliente e da complexidade. Considerando, por exemplo, que temos uma fase para programar, devemos distribuir as

histórias para essa fase em duas *sprints* de duas semanas. Temos dois programadores plenos (entre júnior – menos experiente –, e sênior – mais experiente) e um sênior, o líder. Entre todas as histórias do Product Backlog já priorizadas e estimadas, o líder repassa ao produtor as estimativas das histórias para incluir no cronograma-mestre do projeto.

## Vamos à prática!

Supondo que estamos desenvolvendo a jogabilidade da fase 1 do Alien City com três programadores, conforme citado anteriormente, alocamos os recursos em função da complexidade da tarefa e do nível do programador. Por exemplo, Raphael é sênior, Deivith e Juliano são plenos. A velocidade de Raphael é maior do que a dos outros dois. Em função de Raphael ter mais prática, ele estimou as tarefas listadas no cronograma com base em sua experiência pregressa. As tarefas de programação e arte, conforme a tabela a seguir, tiveram sua duração estimada considerando um prazo pessimista, isto é, no pior caso elas durarão o tempo definido, mas em média serão concluídas mais rapidamente. Vamos ver, então, como ficaram os prazos. Observe a imagem do cronograma-mestre.

| | Task Mode | Task Name | Duration | Start | Finish | Predecessors | Resource Names | Notes |
|---|---|---|---|---|---|---|---|---|
| 1 | | **Produção da Fase 1 Alien City** | **24 days** | **Mon 26/10/15** | **Thu 26/11/15** | | | |
| 2 | | **Programação** | | | | | **Raphael;Juliano** | |
| 3 | | Implementar a Jogabilidade "Impedir Abdução" | | | | | | |
| 4 | | **Programar Eventos** | | | | | | |
| 5 | | **Início Abdução** | | | | | | |
| 6 | | Programar Gatilho em função da localização | 2 days | Tue 03/11/15 | Wed 04/11/15 | | Deivith;Raphael | Programação |
| 7 | | **Salvamento** | | | | | | |
| 8 | | Sensor de proximidade Ray 360° | 3 days | Mon 26/10/15 | Wed 28/10/15 | | Raphael | Programação |
| 9 | | Controles do Jogador | 2 days | Thu 29/10/15 | Fri 30/10/15 | | Raphael | Programação |
| 10 | | **Salvamento Bem Sucedido** | | | | | | |
| 11 | | Disparo de Animação | 4 days | Mon 26/10/15 | Thu 29/10/15 | 18 | Deivith | Programação |
| 12 | | Lógica de Avaliação de Distância+Evento Controle | 1 day | Mon 02/11/15 | Mon 02/11/15 | 8;9 | Juliano | Programação |
| 13 | | **Salvamento Mal Sucedido** | | | | | | |
| 14 | | Disparo de Animação | 2 days | Fri 30/10/15 | Mon 02/11/15 | 19 | Deivith | Programação |
| 15 | | Lógica de Avaliação de Distância+Evento Controle | 1 day | Tue 03/11/15 | Tue 03/11/15 | 8;9 | Juliano | Programação |
| 16 | | **Arte** | | | | | | |
| 17 | | Animação de Abdução | | | | | | |
| 18 | | Animação Salvamento Bem-Sucedido | 3 days | Mon 26/10/15 | Wed 28/10/15 | | | |
| 19 | | Animação Salvamento Malsucedido | 3 days | Mon 26/10/15 | Wed 28/10/15 | | | |

O cronograma-mestre mostra as atividades de programação e arte, seguidas das datas de início e fim de cada tarefa e o responsável por cada uma delas.

Cronograma-mestre

A equipe de programação ficou com as seguintes atividades e estimativas.

- Programação de gatilho em função da localização: tem a duração de dois dias, com início em 03/11/15 e fim em 04/11/15; alocada ao recurso Raphael.
- Sensor de proximidade Ray 360°: tem a duração de três dias, com início em 26/10/15 e fim em 28/10/15; alocada ao recurso Raphael.
- Controles do jogador: tem a duração de dois dias, com início em 29/10/15 e fim em 30/10/15; alocada ao recurso Raphael.
- Salvamento Bem-Sucedido – Disparo de Animação: tem duração de quatro dias, com início em 26/10/15 e fim em 29/10/15; alocada ao recurso Deivith.
- Salvamento Bem-Sucedido – Lógica de Avaliação de Distância + Evento de Controle: tem duração de um dia, com início em 02/11/15 e término no mesmo dia; alocada ao recurso Juliano.
- Salvamento Malsucedido – Disparo de Animação: tem duração de dois dias, com início em 30/10/15 e término em 02/11/15; alocada ao recurso Deivith.
- Salvamento Malsucedido – Lógica de Avaliação de Distância + Evento Controle: tem duração de um dia, com início e fim em 03/11/15; alocada ao recurso Juliano.

As atividades de arte ficaram com as seguintes tarefas e estimativas.

- Animação de abdução: duração não estimada.
- Animação de salvamento bem-sucedido: tem duração de três dias, com início em 26/10/15 e fim em 28/10/15, sem recurso alocado, ainda.
- Animação de salvamento malsucedido: tem duração de três dias, com início em 26/10/15 e fim em 28/10/15, sem recurso alocado, ainda.

## Lembre-se de que as dependências são alocadas a fim de não travarem o desenvolvimento.

Por fim, as estimativas são um “chute” direcionado, pois não se sabe ao certo quanto tempo a equipe levará para completá-las, sabendo que equipes diferentes podem levar tempos diferentes. Vimos que é possível recorrer a experiências em outros projetos ou a tarefas do mesmo projeto para estimarmos o tempo de duração da nossa tarefa. Distribuímos as tarefas ou histórias no time-box de uma sprint. Desta forma, vimos uma forma de controlar melhor a execução e a avaliação da estimativa das tarefas. Com este exemplo apresentado e outros obtidos pelas pesquisas que realizará, você pode começar a calibrar suas habilidades de estimação para evoluí-las com suas experiências.

Não deixe de utilizar a bibliografia sugerida, pois nela há informações importantes sobre como elaborar um cronograma e estimar as tarefas para o projeto. Lembrando, ainda, que a internet é um espaço rico em informações sobre projetos de jogos, com materiais em vídeo e artigos. Pesquise também sobre as estimativas em projetos de *software*.

Passamos pelas importantes etapas do ciclo de produção de programação, o protótipo, o cronograma de atividades e as estimativas, seguimos agora com a programação.

## Referências bibliográficas

PHAM, Andrew X; PHAM, Phuong-van. ***Scrum* em ação:** gerenciamento e desenvolvimento ágil de projetos de *software*. [S.l.]: Novatec. 2011.

CHANDLER, Heather Maxwell. **Manual de produção de jogos digitais**. 2ª ed. Porto Alegre: Bookman, 2012.