

CAPÍTULO 2 SCRUM TEAM

2. Perfil de Dev.

Em uma equipe de desenvolvimento aplicada ao sistema ágil Scrum, há uma prévia comunicação para decisão de cargos. Nessa conversa, por votação, consenso ou demandas superiores, cria-se as divisões de cargo, PO, SM e ST.

Na decisão de escolha, o perfil pessoal de cada um é de grande ajuda, os mais paralelos a líderes e secretários tendem a ocupar cargos diferentes do ST, aqueles que pretendem focar em trabalho, na parte de desenvolvimento, aprimorando suas habilidades técnicas e evoluindo na comunicação de equipe, costumam ser do Scrum Team.

Retomando conceitos da metodologia Scrum, sabe-se que para a estimativa de cada User Story é considerada também o tempo de estudo; é mais comum que os Developers se interessem por essa parte, pois amplia um repertório diretamente técnico. Exemplificando: se a tarefa for de criar uma homepage em um website, caso haja tempo na Sprint, os menos conhecedores do assunto serão encarregados da tarefa, com a supervisão e assistência daqueles com maior habilidade no assunto.

Sabe-se, então, que o perfil ideal de um Developer participante da Equipe Scrum é aquele que aspira ver a equipe avançar e entende que, para isso, deve continuar em constante evolução pessoal: em suas Hard Skills e Soft Skills; a única coisa que não pode ocorrer de maneira alguma é estagnar, pois caso ocorra, isso dificulta o próprio trabalho do Developer e a equipe como um todo, é impossível, no Scrum, a equipe funcionar só. Scrum é como uma máquina, os cargos são as engrenagens que farão a máquina funcionar: ela até consegue desempenhar sua função sem uma das engrenagens, mas só pode funcionar como deveria se todas girarem de acordo.

Fonte: <https://www.mdpi.com/2227-7390/8/9/1553>

2.1. Trabalhos do Scrum Team

É fato que todos os membros numa equipe se ajudam e fazem seus deveres juntamente, porém, há deveres exclusivos também para cada pessoa e para cada cargo. Nesta seção serão tratados alguns trabalhos que apenas o Scrum Team pode desenvolver e finalizar.

Técnico:

Boa parte dos trabalhos técnicos serão de responsabilidade geral, na qual todos da equipe terão que atuar, pelo menos uma vez, como desenvolvedores e aplicarem seus conhecimentos em prol da entrega de um bom/satisfatório produto.

Entretanto, é consenso de que algumas tarefas específicas terão exclusividade na área dos Developers, como resolver os impedimentos técnicos de determinada problemática: deve-se ocorrer a responsabilização do Scrum Team nesse âmbito pois o Scrum Master será responsável pela resolução de impedimentos organizacionais.

Além disso, há a preparação de demonstrações técnicas do produto nas Sprint Reviews, apesar do Product Owner e do SM poderem participar como apresentadores nas Sprint Reviews, o trabalho de descomplicar a demonstração técnica do produto recai sobre o Scrum Team.

Social:

Auto organização interna pode ser colocado juntamente aos trabalhos exclusivos, pois dentre as asks direcionadas ao ST, o grupo tem de distribuir para cada um o que mais se adequa e relaciona, caso contrário, todos terão de estudar mais que o necessário e, apesar de obterem conhecimento, desperdiçarão tempo de desenvolvimento que foi estimado para a Sprint.

A estimativa de alguns objetivos também são exclusivos para o time dos Developers, visto que esses específicos serão desenvolvidos por eles, portanto, não há maneira melhor de definir a estimativa de uma tarefa senão com alguém que sabe a manejar.

Percebe-se, então, que obrigações são existentes e constantes no Scrum, além de se dividirem entre obrigações do grupo, de cargo e pessoais, para fins de organização, otimização do uso de tempo e aspiração para a construção de um produto que satisfaça o cliente.

Fonte: <https://dl.acm.org/doi/abs/10.1145/3202710.3203149>
<https://ieeexplore.ieee.org/abstract/document/6681108>

2.2. Dinâmica em Sprints

Um dos principais pontos do Scrum é sua divisão por Sprints, que são, resumidamente, períodos de trabalho e entregas parciais do produto requisitado pelo cliente. Nessas Sprints, ocorrem alguns episódios: Sprint Planning, Daily Scrum, Sprint Review e Sprint Retrospective; esses tópicos já foram muito abordados, portanto não haverá significativo aprofundamento acerca deles, porém, será comentado sobre o posicionamento do Scrum Team.

Sprint Planning:

Como um dos principais objetivos da Sprint Planning é separar em seções a resolução de um problema raiz. O dever de planejar é do Product Owner, mas o auxílio primordial deve ser do ST, pois, como comentado anteriormente, os

membros têm de estimar juntamente o período a ser concluído determinada atividade, afinal, eles são os que vão trabalhar na maior parte ou todas elas.

Daily Scrum:

Primordialmente, “Dailys” são um dos passos mais relevantes para toda a metodologia Scrum, é nela que se acompanham os progressos e problemáticas do processo. A participação do ST nas Dailys são mais do que importantes, são indispensáveis, eles devem repassar todos seus avanços, pedir ajuda, sugestões, todo e qualquer tipo de comunicação importante deve ser priorizado no dia da Daily, para que não se perca.

Para Devs, manter a Daily como dia para comunicar é muito vantajoso, visto que há tempo para maturar a dúvida/problemática e discutir com o grupo para possíveis soluções e incrementos, além de tudo ser anotado pelos membros da equipe.

Sprint Review:

Ao finalizar uma Sprint, o Scrum Team apresenta o que foi desenvolvido. Essa apresentação é colaborativa, o PO e o SM podem participar e provavelmente irão em algum ponto, mas o protagonismo inicial é do Scrum Team, que demonstra tecnicamente o produto funcionando. O objetivo é coletar feedback e validar se o incremento atende aos critérios definidos, tudo isso com base no que o cliente e avaliador falam (tópicos que serão tragos à tona em futuras Dailys da equipe).

Sprint Retrospective:

Em um evento semelhante à Daily, haverá o Sprint Retrospective, no qual todos olham para trás e percebem o que devem melhorar no projeto, seguindo sugestões de clientes e avaliadores da Sprint Review. Muitas dessas sugestões terão de partir dos Devs, que entendem onde erraram ou o que acertaram, o que podem melhorar e o que não conseguem e preferem manter. Tudo deve ser conversado pois o objetivo é não haver falhas comunicativas.