"estrito" os padrões descritos devem ser mantidos a todo custo. Por isso mesmo ele deve cobrir muitos cenários diferentes.

[00:34] Porque se não tenho nenhuma possibilidade de criar algo novo, tem que seguir à risca o que está nele, ele tem que cobrir todos os cenários possíveis. Então esse modelo "Estrito" tende a ser um pouco mais extenso do que os outros modelos.

"Solto" onde ele tem um uso opcional e dessa forma temos maiores possibilidades para maior experimentação, criar coisas diferentes que não estavam previstas no Design System.

[01:03] Porém, temos que tomar um pouco de cuidado com isso, porque se ele é solto demais, para que existe um Design System, já que não preciso utilizar ele? Então ele fica meio sem sentido. Então tem que ter um equilíbrio entre estrito e variável demais.

[01:18] O terceiro modelo que vou comentar é o "modular", onde nele temos partes reutilizáveis e intercambiáveis, ou seja, dessa forma crio um componente que é para Android, mas pode ser utilizado para outro elemento, para outro produto, para outro dispositivo.

[01:38] Então ele tem essa documentação de elementos que podem variar os seus usos. É por causa disso ele tem maior dificuldade de ser implementado e também é ótimo para produtos extensos.

[01:47] Porque pensa na aula passada que comentei sobre o Spotify que tem até na geladeira. Como uso? Vou ter no meu Design System falando todos os usos da minha interface para geladeira? Pode ser que não, então às vezes posso pegar um elemento que esteja lá documentado para Android e adaptar para a minha realidade.

[02:06] Então esse daqui é o modelo "modular" de Design System. Agora vamos falar sobre diferentes modelos de equipes para trabalhar com esse produto. Começando pela equipe "solitário" onde um time, por exemplo, em uma empresa grande podemos ter diversos times. Um time de Android, outro time de iOS e outro time para sites.

[02:26] Nesse caso do "solitário", o time de Android fez um Design System e aí compartilha depois com o time de iOS, o tipo de sites. Então uma equipe é responsável pela criação e compartilha com as demais equipes.

[02:40] Porém, é necessário cuidado, porque com o passar do tempo, pode ser que não funcione, pois o Design System é feito para suprir as demandas de cada uma das equipes e de cada um dos times.

[02:50] Então se um time só é responsável pela criação dele e depois compartilha com outros times, pode ser que com o tempo não funcione. O "centralizado" é um pouco diferente. Tem um time que é focado especificamente em criar esse Design System e aí ele compartilha com todas as outras equipes.

[03:07] Então a principal atribuição dessa equipe é pegar feedbacks de todos os outros times e por meio desses feedbacks implementar no Design System. Então esse é o modelo "centralizado", que também é um pouco diferente do modelo "federado".

[03:24] Porque no modelo "federado" nós temos várias equipes e aí pesamos: "caraca, aquela pessoa daquela equipe de Android é muito talentosa, aquela outra da equipe de iOS é muito talentosa e aquela outra da outra equipe é muito talentosa".