1

00:00:01,000 --> 00:00:06,000

本中文字幕由自发性 Scrum 团队提供：

PO 周龙鸿／翻译：王可帆,

任兰怡, 江岳龙, 余俊杰,

周玉萍, 林士智, 林清雅, 邱畯丞, 张峰睿, 张越程,

张巨鑫, 陈美凤, 黄久

娟, 廖淑萍／时间轴：ZephyR(西风)

2

00:00:07,100 --> 00:00:08,780

嘿，你回来了，太好了

3

00:00:08,780 --> 00:00:10,940

现在你大概忘记第一集说了什么

4

00:00:10,940 --> 00:00:13,330

让我们快速的回顾一下

5

00:00:13,330 --> 00:00:16,500

我们的文化根基于敏捷原则

6

00:00:16,500 --> 00:00:18,580

所有工程都在小队内进行

7

00:00:18,580 --> 00:00:22,340

我们试着让他们彼此松耦合，同时高度一致

8

00:00:22,340 --> 00:00:27,320

我们喜欢异花传粉，建立了内部开源代码模型

9

00:00:27,320 --> 00:00:32,140

通过解耦，各小队进行小而频繁的发布版本

10

00:00:32,140 --> 00:00:35,580

我们的自助模型使得交接尽量减少

11

00:00:35,580

--> 00:00:40,620

且运用「发布火车」和「特性开关」

使工作成果能尽早且频繁地上线

12

00:00:40,620

--> 00:00:46,940

文化全部是关于人的，因此我们专注在激励、社区和信任上面

13

00:00:46,940 --> 00:00:49,340

而不是结构与控制

14

00:00:49,340 --> 00:00:50,600

以上是第一集的内容

15

00:00:50,600 -->

00:00:52,780

现在，我想谈谈「失败」

16

00:00:53,400 --> 00:00:55,120

我们的创办人Daniel说得好

：

17

00:00:55,120 --> 00:00:58,399

「我们就是要比别人更快地犯错」

18

00:00:58,399 -->

00:01:01,210

是的，我知道，听起来有点疯狂

19

00:01:01,210 --> 00:01:02,220

但是这个想法是这样的

20

00:01:02,230 --> 00:01:05,760

为了要建造真正酷的东西

我们都无可避免地会犯错

21

00:01:05,760 --> 00:01:06,840

对不对？

22

00:01:06,840 --> 00:01:08,720

但是每次失败同时也是一次学习

23

00:01:08,720 --> 00:01:11,520

所以当我们失败时，我们希望快速失败

24

00:01:11,520 -->

00:01:14,530

我们就可以快速学习

然后可以快速改进

25

00:01:14,530 --> 00:01:16,440

这是个为了长期成功的策略

26

00:01:16,440 --> 00:01:18,300

就像小孩子一样

27

00:01:18,300 -->

00:01:21,060

你可放在婴儿摇篮里，她会很安全

28

00:01:21,060 --> 00:01:23,750

但是她学不到什么

，也不会很开心

29

00:01:23,750 --> 00:01:26,470

如果你让她到处跑去探索世界

30

00:01:26,470

--> 00:01:28,700

她有时候会失败，会摔倒

31

00:01:28,700 --> 00:01:33,860

但她会更开心，

发展得更

快，而伤口通常会痊愈的

32

00:01:33,860 --> 00:01:36,289

所以Spotify是一个包容失败的环境

33

00:01:36,289 --> 00:01:41,100

相比避免失败，

我们对快速失败和痊愈更有兴趣

34

00:01:41,140 -->

00:01:44,060

我们内部博客有这样的文章，比如：《庆祝失败》

35

00:01:44,060 --> 00:01:47,000

还有故事，像是：

《我们是如何搬起石头砸自己的脚」

36

00:01:47,000 --> 00:01:48,740

有些小队甚至有「失败墙」

37

00:01:48,740 --> 00:01:51,860

人们在那里展示他们最新的失败与学习成果

38

00:01:51,860 -->

00:01:54,610

没有学习成果的失败，那就只是失败

39

00:01:54,610 --> 00:01:58,170

因此当出问题的时候，我

们通常会进行事后检视

40

00:01:58,170 --> 00:02:00,200

这与「是谁的错？」无关

41

00:02:00,200

--> 00:02:01,750

只关乎「发生了什么？」

42

00:02:01,750 --> 00:02:04,440

「我们学到了什么？」

「我们将如何改变？」

43

00:02:04,440 --> 00:02:07,640

事后检视实际是我们的事故管理流程里的一部分

44

00:02:07,640 --> 00:02:10,950

因此当问题解决后，事故处理单并未结束

45

00:02:10,950

--> 00:02:15,320

只有当我们有所学习收获时才结束，以避免相同问题再发生

46

00:02:15,320 -->

00:02:17,890

修复流程，而不仅仅是产品

47

00:02:17,890 --> 00:02:21,280

附带一提

所有小队每隔几

个星期就会进行回顾会议

48

00:02:21,280 --> 00:02:24,780

来讨论哪些地方做得很好

哪些地方将来还要改善

49

00:02:24,780 --> 00:02:28,170

总而言之， Spotify具备一个有力的文化：

持续改进

50

00:02:28,170 -->

00:02:31,300

自下而上驱动，自上而下支持

51

00:02:31,300 --> 00:02:35,620

然而，失败必须是非致命的

或者我们活着并非为了再次失败

52

00:02:35,620 --> 00:02:38,880

因此我们提倡这观念「有

限爆炸半径」

53

00:02:38,880 --> 00:02:40,860

这架构是相当解耦的

54

00:02:40,860 -->

00:02:42,360

因此当一个小队犯错了

55

00:02:42,360 --> 00:02:46,580

它通常只会影响系统的一小部分

，而不是搞垮所有

56

00:02:46,580 --> 00:02:51,000

且因小队对端到端负责，无需交接切换

57

00:02:51,000 --> 00:02:53,240

他们通常可以快速修复问题

58

00:02:53,240 -->

00:02:55,740

而大多数的新功能逐步发布，会先发布给少量的

使用者来测试

59

00:02:55,740 --> 00:02:59,220

并且作密切的观察

60

00:02:59,220 --> 00:03:03,400

当这些功能确认稳定了之后

我们会逐步地发

布给世界各地的使用者

61

00:03:03,420 --> 00:03:04,700

所以就算中间出了状况

62

00:03:04,700

--> 00:03:11,400

通常也只会影响到系统的一小部分及少量用户

而且影响的时间很短暂

63

00:03:11,450

--> 00:03:13,880

这种有限的损害范围让小队勇于

64

00:03:13,880 --> 00:03:17,210

做更多的小型

试验，并加速学习的脚步

65

00:03:17,210 --> 00:03:20,900

而不是浪费时间在风险的预测及控制

66

00:03:20,980 --> 00:03:22,700

Mario Andretti说得好：

67

00:03:22,700 -->

00:03:26,130

「如果事事尽在掌握

　那你走得就太慢了」

68

00:03:26,130 --> 00:03:29,110

好，

那现在让我们来谈谈产品开发吧

69

00:03:29,110 --> 00:03:32,640

我们的产品开发方法基于

「精益创业」原则

70

00:03:32,640 --> 00:03:33,959

简单来说就是

71

00:03:33,959 -->

00:03:37,540

「思考、构建、交付、调整」

72

00:03:37,540 --> 00:03:40,450

产品开发最大的风险

就在产出错误的产品

73

00:03:40,450 --> 00:03:43,819

所以在决定构建新产品或主要功能前

74

00:03:43,819 --> 00:03:45,850

我们会先研究

75

00:03:45,850 --> 00:03:49,760

我们的客户是不是

真的想要这个功能？

这是否能帮大家解决真正的问题？

76

00:03:49,760 --> 00:03:54,980

接着我们

会写一个类似新闻稿或电梯演讲

的产品叙述来展示其收益

77

00:03:54,980 -->

00:03:58,459

比如说：「可以保存的电台！」

或「关注你最喜欢的歌手」这类的产品特性标语

78

00:03:58,459 -->

00:04:00,380

我们同时也会做一些假设，像是

79

00:04:00,380 --> 00:04:03,680

在我们的核心指标中，

这个功能会如何影响用户的行为

80

00:04:03,680 --> 00:04:06,650

他们是否会分享更多音乐？

他们会更常来登入使用吗？

81

00:04:06,650 --> 00:04:09,450

为此我们建立了各种原型让大家试用

82

00:04:09,450 --> 00:04:13,100

让他们体验新功能的同时获取他们的感受和反应

83

00:04:13,160 -->

00:04:15,650

一旦我们确定这功能或产品值得构建时

84

00:04:15,650 --> 00:04:19,870

我们会开始

打造MVP，也就是

「最小可行性产品」

85

00:04:19,870 --> 00:04:23,460

刚刚好去满足之前的产品叙述

但离完整的功能还很远

86

00:04:23,460 --> 00:04:26,370

你也可以称它是「

刚好受欢迎产品」

87

00:04:26,370 --> 00:04:29,919

当我们将一些功能上线时，会开始下一个

学习阶段

88

00:04:29,919 --> 00:04:32,100

因此我们会尽快的完成

89

00:04:32,100 -->

00:04:35,260

我们仅向部分使用者发布MVP

90

00:04:35,260 --> 00:04:39,350

并使用像是A/B测试的技

术

来衡量影响及验证假设

91

00:04:39,350 --> 00:04:42,550

同时，小队会来监控数据

持续调整产品和重新部署

92

00:04:42,550 --> 00:04:44,590

直到达到预期的影响

93

00:04:44,590 -->

00:04:46,710

接着他们才会逐步发布给世界上其他使用者

94

00:04:46,710 --> 00:04:49,090

当然还得花点时

间解决一些实际问题

95

00:04:49,090 --> 00:04:51,570

例如运营及扩展

96

00:04:51,570 --> 00:04:54,010

等到产品或功能全面推出的时候

97

00:04:54,010 --> 00:04:55,820

我们就知道它是一个成功的产品

98

00:04:55,820 --> 00:04:58,290

因为如果不是，我们就不会推出

99

00:04:58,290 --> 00:05:00,520

影响总是比速度更重要

100

00:05:00,520 --> 00:05:05,040

只有当预期影响达成时

产品特性才算是完成

101

00:05:05,040 --> 00:05:08,610

与这个影片中很多东西一样，这是我们尝试的工作方式

102

00:05:08,610 --> 00:05:12,360

但实际上并非一帆风顺

103

00:05:12,360 --> 00:05:16,430

当处于实验阶段时

我们如何做计划？

104

00:05:16,430

--> 00:05:18,770

我们怎么知道什么时候会发布什么产品？

105

00:05:18,770 --> 00:05:21,590

嗯，

说老实话，我们自己也没个准

106

00:05:21,590 --> 00:05:24,220

比起预测，我们更关心创新

107

00:05:24,220 --> 00:05:28,420

100%的可预测性代表着0%的创新

108

00:05:28,420 --> 00:05:30,920

我们大概在天平的

109

00:05:31,300 --> 00:05:32,160

这

里

110

00:05:32,160 --> 00:05:35,000

当然，有时候我们需要做出交付的承诺

111

00:05:35,010

--> 00:05:37,160

以配合合作伙伴的整合或市场营销活动

112

00:05:37,160 --> 00:05:40,820

这常牵涉到

标准的敏捷计划技术

113

00:05:40,820 --> 00:05:43,020

像是速度及燃烧图

114

00:05:43,020

--> 00:05:46,280

但如果我们一定要给出一个期限

我们通常会延迟承诺

115

00:05:46,280

--> 00:05:48,840

直到产品特性都已经被证实且几近完成

我们才会给承诺

116

00:05:48,840 -->

00:05:53,150

通过降低预测性的要求，小队会关注交付价值

117

00:05:53,150 --> 00:05:56,669

而不是成为某人武断计划的奴隶

118

00:05:56,669 -->

00:05:57,770

有位产品负责人曾说：

119

00:05:57,770 --> 00:06:00,880

「我觉得我的小队就像一群

志愿者

120

00:06:00,880 --> 00:06:03,820

聚集在此从事他们所热衷的事」

121

00:06:03,820 -->

00:06:06,490

所以，创意何来？

122

00:06:06,490 --> 00:06:10,400

惊人的新产品始于人

123

00:06:10,400 --> 00:06:11,820

与灵感的结合

124

00:06:11,820 --> 00:06:16,400

而唯有允许人们玩耍与尝

试，创意才会成真

125

00:06:16,419 --> 00:06:22,000

所以我们鼓励每个人花10%的时间参与

黑客日

或黑客周

126

00:06:22,060 --> 00:06:25,110

在这段期间大家根据自己的想法尽情去实验或创作

127

00:06:25,110

--> 00:06:27,320

就像「拨一首歌」这个产品

128

00:06:27,320 --> 00:06:30,300

只要拿起电话并拨

打你想听得那首歌的号码

129

00:06:30,320 --> 00:06:31,720

这有用吗？

130

00:06:31,720 -->

00:06:32,440

不重要

131

00:06:32,440 --> 00:06:37,340

重点是，只要你试的点子够多

总会有挖到

宝的时候

132

00:06:37,340 --> 00:06:40,840

往往，做黑客时所学到的知识比黑客

这个行为本身要来

得有意义

133

00:06:40,840 --> 00:06:42,470

除此之外，也挺好玩的

134

00:06:42,470 -->

00:06:45,639

所以我们每年举办二次Spotify内部黑客周

135

00:06:45,639 --> 00:06:48,350

好几百

人整周都在闭关当黑客

136

00:06:48,350 --> 00:06:51,020

我们的头号标语就是「令酷成真」

137

00:06:51,020 --> 00:06:54,990

用你想要的方式和你想要的人

共同创作你想要的事物

138

00:06:54,990 --> 00:06:57,110

然后在周五的大型展示会中展示

139

00:06:57,110 -->

00:06:58,390

我们总是惊讶的发现

140

00:06:58,390 --> 00:07:02,940

我们可以在一周内

透过这种

自由创作搞出不少名堂

141

00:07:02,940 --> 00:07:05,040

不论是用棒棒糖做成的直升机

142

00:07:05,040 --> 00:07:06,770

或以全新的方式发现音乐

143

00:07:06,770 --> 00:07:10,610

这说明

创新并不是那么难

144

00:07:10,610 --> 00:07:11,880

人是天生的创造者

145

00:07:11,880 -->

00:07:14,990

所以就完全放手让他们去尝试

146

00:07:14,990 --> 00:07:18,199

整体而言

，我们的文化是鼓励实验的

147

00:07:18,199 --> 00:07:21,179

举例来说，我们该用工具A还是工具B

？

148

00:07:21,180 --> 00:07:21,760

不清楚

149

00:07:21,760 --> 00:07:23,480

让我们两者都尝试

并且比较一下结果

150

00:07:23,480 --> 00:07:26,340

或者我们是不是真的需要Sprint计划会议？

151

00:07:26,340 --> 00:07:27,000

不清楚

152

00:07:27,000 --> 00:07:29,440

那我们就跳过一些

会议来看是否真的需要

153

00:07:29,440 --> 00:07:33,240

或我们该在歌手页放5或10首排行榜歌曲？

154

00:07:33,240 --> 00:07:33,740

不清楚

155

00:07:33,740 --> 00:07:35,860

让我们两种都试看

看并且评估成效

156

00:07:35,870 --> 00:07:38,800

一开始Spotify黑客周只是个试验

157

00:07:38,800 --> 00:07:40,840

现在它已成为我们文化的一部分

158

00:07:40,840 --> 00:07:42,410

所以

，与其在那边争论个你死我活

159

00:07:42,410 --> 00:07:45,600

不如好好的来讨论，比如“有哪些假设？”

160

00:07:45,600 --> 00:07:48,080

“我们学到了什么？接下来又将尝试什么？”

161

00:07:48,080 -->

00:07:52,310

这让我们的决定多是基于客观数据的

而不是来自个人主观意见

162

00:07:52,310 -->

00:07:54,750

或基于职权的

163

00:07:54,750 --> 00:07:58,190

虽然我们乐于尝试，并且试着用不同

方式做事

164

00:07:58,190 --> 00:08:00,460

但我们的文化非常排斥浪费

165

00:08:00,460 -->

00:08:03,139

大家很快停止作任何没有附加价值的事

166

00:08:03,139 --> 00:08:06,490

如果有效就

继续做，反之，就停止

167

00:08:06,490 --> 00:08:09,340

举例来说，到目前为止

有些事是对我们有

帮助的

168

00:08:09,340 --> 00:08:15,300

像回顾会议、每日站立会议、

Google文档、Git及公会的研讨会议

169

00:08:15,310 --> 00:08:19,330

有些事情对我们没用，像是：时间报告和交接

170

00:08:19,330 --> 00:08:22,759

独立的测试团队或测试阶段

以及工作任务估算

171

00:08:22,759 -->

00:08:24,620

我们几乎不做这些事

172

00:08:24,620 --> 00:08:27,580

我们也对无效的会议

173

00:08:27,580 --> 00:08:30,990

以及一些企业官话避之而唯恐不及

174

00:08:30,990 -->

00:08:34,940

最常见的资源浪费就是俗称的大型项目

175

00:08:34,940 --> 00:08:39,920

基本上就是

那种要很多小队

一起密切合作数个月的项目

176

00:08:39,920 --> 00:08:42,020

大型项目意味着巨大

的风险

177

00:08:42,029 --> 00:08:44,140

所以我们力求短小精干

精简需求

178

00:08:44,140

--> 00:08:48,280

并且努力把项目拆解成一系列小的工作

179

00:08:48,290 --> 00:08:52,119

然而有

些时候大型项目也是合理的

180

00:08:52,119 --> 00:08:54,589

且潜在利益远胜于风险

181

00:08:54,589 --> 00:08:58,069

在那些情况，我们发现

有些实际的作法是很必要的

182

00:08:58,069

--> 00:09:03,040

用物理或电子白板的各种组合来可视化进展

183

00:09:03,040 --> 00:09:04,720

进

行每日同步会议

184

00:09:04,720 --> 00:09:08,040

所有小队共同参与讨论，解决相互的依赖

185

00:09:08,040 --> 00:09:11,760

每一或两个星期进行演示会议

186

00:09:11,760 -->

00:09:15,200

将产品各部分整合在一起，然后让利益干系人对整体进行评估

187

00:09:15,200 -->

00:09:21,069

这些实践降低了风险和浪费，因为通过短而快速

的反馈循环，逐步改善了协作方式

188

00:09:21,069 --> 00:09:24,649

我们也发现到一个项目需要

一个小而紧凑的领导型团队

189

00:09:24,649 --> 00:09:26,720

来持续观察整体大局

190

00:09:26,720 -->

00:09:31,399

通常我们会有一位技术主管，一位产品主管

有时还会有一位设计主管，三者通力合作

191

00:09:31,399 --> 00:09:37,369

总之，对于怎样来运作大型项目，我们还在进行许多尝试

毕竟我们还不是很擅长

192

00:09:37,369 --> 00:09:40,629

有一件事情我们在全力应付，那就是成长的烦恼

193

00:09:40,629

--> 00:09:44,089

当我们成长时，我们面临着陷入混乱的风险

194

00:09:44,089 --> 00:09:47,550

但若

我们过度补偿，去增加太多的结构和流程

195

00:09:47,550 --> 00:09:50,300

我们又会陷入官僚化的

风险

196

00:09:50,300 --> 00:09:51,379

那就更糟糕了

197

00:09:51,379 --> 00:09:55,740

所以

真正关键的问题是什么样的

「最小可行性官僚系统」？

198

00:09:55,740 --> 00:09:58,910

可以让我

们使用最少的结构和流程来

199

00:09:58,910 --> 00:10:01,079

避免陷入完全混乱

200

00:10:01,079

--> 00:10:04,079

这两边都会以造成浪费

201

00:10:04,079 --> 00:10:08,399

所以排除浪

费的文化与敏捷的思维能

帮助我们保持平衡

202

00:10:08,399 --> 00:10:12,790

减少浪费的关键在于

可视化和经常沟通

203

00:10:12,790 --> 00:10:15,270

因此，除了回顾

会议和事后检视之外

204

00:10:15,280 --> 00:10:18,400

许多由小队与部落

会采用大型可视化的「改进清单」

205

00:10:18,400 --> 00:10:21,920

显示哪些事件阻碍了我们

以及我们正在处理哪些障碍

206

00:10:21,920 --> 00:10:24,819

我们也喜欢讨论「牛逼的定义」

207

00:10:24,819 --> 00:10:27,739

举例来说，对这个小队来说

「牛逼的定义」表示

208

00:10:27,739 --> 00:10:31,079

实际完成了事情、让新成员容易跟上

209

00:10:31,080 -->

00:10:33,520

也没有发生反复出现的任务或缺陷

210

00:10:33,520 --> 00:10:36,360

而我们对一个「牛逼的架构」

的定义则包括

211

00:10:36,360 --> 00:10:39,560

我可以在一周内构建、测试

及上线我的功能

212

00:10:39,560 --> 00:10:44,060

我从数据中学习经验

并在第二个星期将改进后的版本上线

213

00:10:44,069 --> 00:10:46,949

记住，「牛逼」

是一个方向，不是一个位置

214

00:10:46,949 --> 00:10:49,020

所以不需要是现实的内容

215

00:10:49,020 --> 00:10:51,320

但是

当我们都同意「牛逼」

应该长什么样子

216

00:10:51,320 --> 00:10:55,440

就有助于我们聚焦在

改善效果和追踪进展

217

00:10:55,440 --> 00:10:57,459

这里有一个改进追踪板的案例

218

00:10:57,459 --> 00:11:00,869

受到一种叫做丰田管理(Toyota Kata)的

精益技术的启发

219

00:11:00,869 --> 00:11:06,140

左上角显示现状

在这个案例中，小队面临质量问题

220

00:11:06,140

--> 00:11:08,380

左下角显示「牛逼的定义」

221

00:11:08,399 --> 00:11:11,959

理想中，我们不会

有质量问题

222

00:11:11,959 --> 00:11:14,309

右上角是现实的目标情形

223

00:11:14,309 -->

00:11:17,829

如果我们靠近「牛逼」一步，

那看起来会是怎样的？

224

00:11:17,829 -->

00:11:23,989

最后，右下角展示了接下来三个让我们

前往目标情形的具体行动

225

00:11:23,989 -->

00:11:27,380

当这些完成时，小队会填上新的行动

226

00:11:27,389 --> 00:11:29,369

这种在小队工作室墙上的动态白板

227

00:11:29,369 --> 00:11:31,759

通常会在下次的回顾会议进行跟进

228

00:11:31,759 --> 00:11:38,000

好，我知道这个影片也许让Spotify的

每件事看起来很

美好

229

00:11:38,050 --> 00:11:41,730

但是，事实上

我们还有许多问题需要处理

230

00:11:41,730 --> 00:11:44,940

我甚至可以开出一个写满痛点的长长的清单

231

00:11:44,950 --> 00:11:47,480

但是我不会，因为这份清单上

的问题会很快过时

232

00:11:47,480 --> 00:11:49,110

我们快速成长，快速改变

233

00:11:49,110 --> 00:11:54,300

今天看起来聪明的解决方案

可能明天会造成一个难缠的新

问题

234

00:11:54,329 --> 00:11:56,679

只因为我们成长了，然后每件事情都不同了

235

00:11:56,679 --> 00:12:01,400

然而，大多数问题都很短暂

因为人们确实会去解决问题

236

00:12:01,480 --> 00:12:05,410

这间公司特别擅长改造

像是架构、流程、组织

237

00:12:05,410 -->

00:12:07,699

或任何必须用来解决问题的事物

238

00:12:07,699 --> 00:12:09,660

而这就是关键所在

239

00:12:09,660 --> 00:12:13,290

健康的文化可修复有问题的流程

240

00:12:13,290 -->

00:12:17,000

由于文化是如此重要

因此我们花了很大的精力去强化它

241

00:12:17,019 -->

00:12:19,259

这支影片只是一个小例子

242

00:12:19,259 --> 00:12:21,110

没有人可以真的拥有文化

243

00:12:21,110 --> 00:12:23,540

但我们有很多成员还是非常重视

244

00:12:23,540 -->

00:12:27,829

比如人力资源团队

以及30位左右的敏捷教练

245

00:12:27,829 -->

00:12:29,629

分散在所有的小队中

246

00:12:29,629 --> 00:12:31,049

同时我们也会执行新员工训练营

247

00:12:31,049 --> 00:12:35,230

由新招募的成员组成临时的小队

然后去实际解决问题

248

00:12:35,230 --> 00:12:39,860

同时也学习我们的测试技术与过程

以及团队协作

249

00:12:39,860

--> 00:12:40,749

以上所有都要在一周内完成

250

00:12:40,749 --> 00:12:43,689

这种紧张而有趣的训练

能让你真正地去融入文化

251

00:12:43,689 --> 00:12:47,179

他们大多会让程序在一周

内上线

令人印象非常深刻

252

00:12:47,179 --> 00:12:51,209

再次强调，失败绝对没关系

只要大家能从中有所学习收获

253

00:12:51,209 --> 00:12:54,809

然后，文化主要还得

透过讲故事来传递

254

00:12:54,809 -->

00:12:59,029

不论在博客，事后检视

演示会议或午餐时

255

00:12:59,029 --> 00:13:02,879

只要

我们彼此持续分享

成功和失败所学到的教训

256

00:13:02,879 --> 00:13:04,910

我认为我们可以一直

保持

这种健康的文化

257

00:13:04,910 --> 00:13:06,970

最后

258

00:13:06,970 -->

00:13:12,739

组织文化实际上就是

每个人的态度和行动的总和

259

00:13:12,739 --> 00:13:17,309

你就是文化

所以去塑造你想要的行为

260

00:13:17,309 --> 00:13:18,799

就这样，说完了

261

00:13:18,799 --> 00:13:20,069

我希望大家喜欢这个故事

262

00:13:20,069 --> 00:13:21,880

感谢

收看

263

00:13:25,040 --> 00:13:26,000

字幕版本：20150915.3